(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2004-326801
(P2004-326801A)
(43)公開日　平成16年11月18日(2004.11.18)

(51)Int.Cl.[7]	F I		テーマコード（参考）
G06F 12/00	G06F 12/00	514E	5B065
G06F 3/06	G06F 3/06	301K	5B082
	G06F 3/06	304F	

審査請求 有　請求項の数 55　ＯＬ　（全 39 頁）

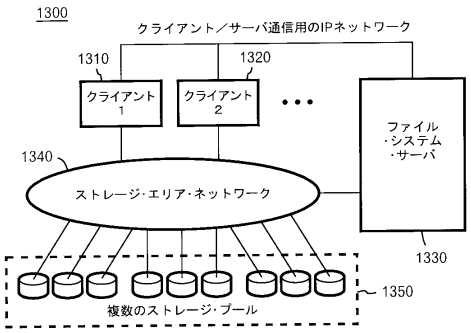
(21)出願番号　特願2004-129471 (P2004-129471)
(22)出願日　平成16年4月26日 (2004.4.26)
(31)優先権主張番号　10/427403
(32)優先日　平成15年4月29日 (2003.4.29)
(33)優先権主張国　米国 (US)

(71)出願人　390009531
インターナショナル・ビジネス・マシーンズ・コーポレーション
ＩＮＴＥＲＮＡＴＩＯＮＡＬ　ＢＵＳＩＮＥＳＳ　ＭＡＳＣＨＩＮＥＳ　ＣＯＲＰＯＲＡＴＩＯＮ
アメリカ合衆国１０５０４　ニューヨーク州　アーモンク　ニュー　オーチャード　ロード
(74)代理人　100086243
弁理士　坂口　博
(74)代理人　100091568
弁理士　市位　嘉宏
(74)代理人　100108501
弁理士　上野　剛史

最終頁に続く

(54)【発明の名称】ファイルのコピー・オン・ライトを実施する方法、システム、およびコンピュータ・プログラム

(57)【要約】　（修正有）
【課題】コンピューティング環境のさまざまなコピー・オン・ライト実施形態を提示すること。
【解決手段】あるコピー・オン・ライト実施形態に、修正のために物理記憶装置からコピー・オン・ライトされるファイルのデータのブロックを読み取るのに使用される第１仮想ブロック－物理ブロック・マッピングを実行するのに読取マッピング・テーブルを使用することと、ファイル・データの修正されたブロックを物理記憶装置に書き込むのに使用される第２仮想ブロック－物理ブロック・マッピングを実行するのに、異なる書込マッピング・テーブルを使用することとが含まれ、ここで、データのブロックのコピー・オン・ライトが、単一の書込動作を使用して達成される。
【選択図】図１３



【特許請求の範囲】
【請求項１】
コンピューティング環境でコピー・オン・ライトを実施する方法であって、
（ｉ）ファイルのデータのブロックを修正のために物理記憶装置から読み取るのに使用される第１仮想ブロック－物理ブロック・マッピングを実行するのに第１マッピング・テーブルを使用することと、
（ｉｉ）前記ファイルの前記データの修正されたブロックを物理記憶装置に書き込むのに使用される第２仮想ブロック－物理ブロック・マッピングを実行するのに第２マッピング・テーブルを使用することであって、データの前記ブロックのコピー・オン・ライトが、単一の書込動作を使用して達成される、前記使用することと
を含む方法。
【請求項２】
前記第１マッピング・テーブルが、読取マッピング・テーブルを含み、前記第２マッピング・テーブルが、書込マッピング・テーブルを含み、前記書込マッピング・テーブルが、前記読取マッピング・テーブルの対応する仮想ブロック－物理ブロック・マッピングと異なる、前記物理記憶装置の物理ブロックにマッピングされる少なくとも１つの仮想ブロックを含む、請求項１に記載の方法。
【請求項３】
前記コピー・オン・ライト実施形態が、さらに、まず修正が部分ブロック書込または全ブロック書込のどちらを含むかを判定することと、部分ブロック書込の場合に、前記使用すること（ｉ）および前記使用すること（ｉｉ）を実行し、そうでない場合に前記使用すること（ｉ）を実行せずに前記実行すること（ｉｉ）を実行することとを含む、請求項１に記載の方法。
【請求項４】
前記実行すること（ｉ）が、データの前記ブロックを物理記憶装置からバッファに読み取ることを含み、前記方法が、さらに、前記使用すること（ｉｉ）を実行する前に前記バッファ内のデータの前記ブロックを修正することを含む、請求項１に記載の方法。
【請求項５】
前記コンピューティング環境が、ファイルシステム・サーバおよび少なくとも１つのクライアントを含むクライアント・サーバ環境を含み、前記使用すること（ｉ）および前記使用すること（ｉｉ）が、前記クライアント・サーバ環境の前記少なくとも１つのクライアントによって実行される、請求項１に記載の方法。
【請求項６】
前記少なくとも１つのクライアントによって実行される前記使用すること（ｉ）および前記使用すること（ｉｉ）が、前記ファイルのデータの前記ブロックのコピー・オン・ライトを実行する時に、前記第１マッピング・テーブルおよび前記第２マッピング・テーブルの少なくとも１つを入手するために前記ファイルシステム・サーバへの少なくとも１つの呼出しを行うことをさらに含む、請求項５に記載の方法。
【請求項７】
データの前記修正されたブロックを物理記憶装置に書き込んだ後に、前記第１マッピング・テーブルを更新することをさらに含み、前記更新することが、前記第１マッピング・テーブルの少なくとも１つの仮想ブロック－物理ブロック変換を、前記第２マッピング・テーブルの対応する仮想ブロック－物理ブロック変換と一致するように修正することを含む、請求項５に記載の方法。
【請求項８】
前記コンピューティング環境が、コンピューティング・ユニットおよび外部記憶ユニットを含み、前記外部記憶ユニットが、前記物理記憶装置を含み、前記使用すること（ｉ）および前記使用すること（ｉｉ）が、前記コンピューティング・ユニットによって実行される、請求項１に記載の方法。
【請求項９】

クライアント・サーバ・コンピューティング環境でコピー・オン・ライトを容易にする方法であって、
前記クライアント・サーバ・コンピューティング環境のファイルシステム・サーバで、ファイルに関する読取マッピング・テーブルおよび書込マッピング・テーブルを維持すること
を含み、前記読取マッピング・テーブルが、修正のために物理記憶装置から前記ファイルのデータのブロックを読み取るのに使用される第１仮想ブロック－物理ブロック・マッピングを実行するのに使用可能であり、前記書込マッピング・テーブルが、前記ファイルの前記データの修正されたブロックを物理記憶装置に書き込むのに使用される第２仮想ブロック－物理ブロック・マッピングを実行するのに使用可能であり、前記読取マッピング・テーブルおよび前記書込マッピング・テーブルを使用して、データのブロックのコピー・オン・ライトが単一の書込動作を使用して達成可能である
方法。
【請求項１０】
前記ファイルに関する前記書込マッピング・テーブルが、前記読取マッピング・テーブルの対応する仮想ブロック－物理ブロック変換と異なる、前記物理記憶装置の物理ブロックにマッピングされる少なくとも１つの仮想ブロックを含む、請求項９に記載の方法。
【請求項１１】
前記ファイルのデータのブロックのコピー・オン・ライトが実行された後に前記読取マッピング・テーブルを更新することをさらに含み、前記更新することが、前記読取マッピング・テーブルの少なくとも１つの仮想ブロック－物理ブロック変換を、前記書込マッピング・テーブルの対応する仮想ブロック－物理ブロック変換と一致するように修正することを含む、請求項９に記載の方法。
【請求項１２】
複数のクライアントを有するクライアント・サーバ環境内でファイルのコピー・オン・ライトを実施する方法であって、
前記クライアント・サーバ環境の複数のクライアントを使用してファイルのコピー・オン・ライトを実行することであって、
（ｉ）前記複数のクライアントの第１クライアントによって、前記ファイルのデータの少なくとも１つのブロックのコピー・オン・ライトを実行することと、
（ｉｉ）前記複数のクライアントの第２クライアントによって、前記ファイルのデータの少なくとも１つの他のブロックのコピー・オン・ライトを実行することと
を含む、実行すること
を含む方法。
【請求項１３】
前記実行すること（ｉ）が、前記第１クライアントによって、単一の書込動作を使用して前記ファイルのデータの前記少なくとも１つのブロックのコピー・オン・ライトを実行することを含み、前記実行すること（ｉｉ）が、前記第２クライアントによって、単一の書込動作を使用してデータの前記少なくとも１つの他のブロックのコピー・オン・ライトを実行することを含む、請求項１２に記載の方法。
【請求項１４】
前記クライアント・サーバ環境のファイルシステム・サーバが、前記ファイルを含む少なくとも１つの共用記憶装置に関連し、前記ファイルシステム・サーバが、前記少なくとも１つの共用記憶装置に保管された前記ファイルに関するマッピング・テーブルを維持し、前記実行すること（ｉ）が、前記第１クライアントによって、前記ファイルシステム・サーバから前記ファイルに関する読取マッピング・テーブルおよび書込マッピング・テーブルを入手することと、前記ファイルのデータの前記少なくとも１つのブロックの前記コピー・アンド・ライトを実行する際に前記読取マッピング・テーブルおよび前記書込マッピング・テーブルを使用することとを含み、前記実行すること（ｉｉ）が、前記第２クライアントによって、前記ファイルシステム・サーバから前記ファイルに関する前記読取マ

ッピング・テーブルおよび前記書込マッピング・テーブルを入手することと、前記ファイルのデータの前記少なくとも１つの他のブロックの前記コピー・アンド・ライトを実行する際に前記読取マッピング・テーブルおよび前記書込マッピング・テーブルを使用することとを含む、請求項１３に記載の方法。

【請求項１５】

前記第１クライアントによって、前記第１クライアントが前記ファイルのデータの前記少なくとも１つのブロックのコピー・オン・ライトを実行したことを前記ファイルシステム・サーバに知らせることと、それに応答して、前記ファイルシステム・サーバによって維持される前記ファイルに関する前記読取マッピング・テーブルおよび前記書込マッピング・テーブルの少なくとも１つを更新することとをさらに含む、請求項１４に記載の方法。

【請求項１６】

前記ファイルシステム・サーバが、前記第１クライアントによってコピー・オン・ライト更新された前記ファイルのデータの前記少なくとも１つのブロックおよび前記第２クライアントによってコピー・オン・ライト更新された前記ファイルのデータの前記少なくとも１つの他のブロックに対するコピー・オン・ライトを前記複数のクライアントのすべてのクライアントが行えなくする、請求項１４に記載の方法。

【請求項１７】

前記クライアント・サーバ環境のファイルシステム・サーバが、前記ファイルを含む少なくとも１つの共用記憶装置に関連し、前記方法が、さらに、前記ファイルの前記コピー・オン・ライトを開始するために前記ファイルシステム・サーバによって前記ファイルを制御することを含み、前記開始することが、前記実行すること（ｉ）および前記実行すること（ｉｉ）によって使用される前記ファイルに関する書込マッピング・テーブルを更新することを含む、請求項１２に記載の方法。

【請求項１８】

クライアント・サーバ環境内でファイルのコピー・オン・ライトを容易にする方法であって、

前記クライアント・サーバ環境の共用記憶ユニットに保管されたファイルのコピー・オン・ライトの実施をファイルシステム・サーバから制御することであって、前記制御することが、前記クライアント・サーバ環境の第１クライアントが前記ファイル内のデータの部分をコピー・オン・ライトできるようにすることと、前記クライアント・サーバ環境の第２クライアントが前記ファイル内の前記データの異なる部分をコピー・オン・ライトできるようにすることとを含み、前記ファイルシステム・サーバが、前記ファイルの分散コピー・オン・ライトの実行を制御すること

を含む方法。

【請求項１９】

前記制御することが、前記ファイルのコピー・オン・ライトを開始するために前記ファイルシステム・サーバによって前記ファイルを制御することを含み、前記開始することが、前記コピー・オン・ライトを実行するのに使用される前記ファイルに関する少なくとも１つのマッピング・テーブルを更新することを含む、請求項１８に記載の方法。

【請求項２０】

前記ファイルシステム・サーバで、前記ファイルに関する読取マッピング・テーブルおよび書込マッピング・テーブルを維持することをさらに含み、前記読取マッピング・テーブルおよび前記書込マッピング・テーブルが、前記コピー・オン・ライトの実行に使用される、請求項１８に記載の方法。

【請求項２１】

前記コピー・オン・ライトの実行の後に、前記ファイルに関する前記読取マッピング・テーブルおよび前記書込マッピング・テーブルの少なくとも１つを更新することをさらに含む、請求項２０に記載の方法。

【請求項２２】

前記制御することが、前記ファイルシステム・サーバによって、前記ファイルの前記コピー・オン・ライトの一部としての、前記第１クライアントによってコピー・オン・ライト更新された前記ファイル内の前記データの前記部分に対する追加更新および前記第２クライアントによってコピー・オン・ライト更新された前記ファイル内の前記データの前記異なる部分に対する追加更新を防ぐことをさらに含む、請求項１８に記載の方法。

【請求項２３】

コンピューティング環境でコピー・オン・ライトを実施するシステムであって、

（ｉ）ファイルのデータのブロックを修正のために物理記憶装置から読み取るのに使用される第１仮想ブロック－物理ブロック・マッピングを実行するのに第１マッピング・テーブルを使用する手段と、

（ｉｉ）前記ファイルの前記データの修正されたブロックを物理記憶装置に書き込むのに使用される第２仮想ブロック－物理ブロック・マッピングを実行するのに第２マッピング・テーブルを使用する手段であって、データの前記ブロックのコピー・オン・ライトが、単一の書込動作を使用して達成される、使用する手段と

を含むシステム。

【請求項２４】

前記第１マッピング・テーブルが、読取マッピング・テーブルを含み、前記第２マッピング・テーブルが、書込マッピング・テーブルを含み、前記書込マッピング・テーブルが、前記読取マッピング・テーブルの対応する仮想ブロック－物理ブロック・マッピングと異なる、前記物理記憶装置の物理ブロックにマッピングされる少なくとも１つの仮想ブロックを含む、請求項２３に記載のシステム。

【請求項２５】

前記コピー・オン・ライト実施形態が、さらに、まず修正が部分ブロック書込または全ブロック書込のどちらを含むかを判定し、部分ブロック書込の場合に、前記使用すること（ｉ）および前記使用すること（ｉｉ）を実行し、そうでない場合に前記使用すること（ｉ）を実行せずに前記実行すること（ｉｉ）を実行する手段を含む、請求項２３に記載のシステム。

【請求項２６】

前記実行する手段（ｉ）が、データの前記ブロックを物理記憶装置からバッファに読み取る手段を含み、前記システムが、さらに、前記使用すること（ｉｉ）を実行する前に前記バッファ内の前記ブロックを修正する手段を含む、請求項２３に記載のシステム。

【請求項２７】

前記コンピューティング環境が、ファイルシステム・サーバおよび少なくとも１つのクライアントを含むクライアント・サーバ環境を含み、前記使用する手段（ｉ）および前記使用する手段（ｉｉ）が、前記クライアント・サーバ環境の前記少なくとも１つのクライアントによって実行される、請求項２３に記載のシステム。

【請求項２８】

前記少なくとも１つのクライアントによって実行される前記使用する手段（ｉ）および前記使用する手段（ｉｉ）が、前記ファイルのデータの前記ブロックのコピー・オン・ライトを実行する時に、前記第１マッピング・テーブルおよび前記第２マッピング・テーブルの少なくとも１つを入手するために前記ファイルシステム・サーバへの少なくとも１つの呼出しを行う手段をさらに含む、請求項２７に記載のシステム。

【請求項２９】

データの前記修正されたブロックを物理記憶装置に書き込んだ後に、前記第１マッピング・テーブルを更新する手段をさらに含み、前記更新する手段が、前記第１マッピング・テーブルの少なくとも１つの仮想ブロック－物理ブロック変換を、前記第２マッピング・テーブルの対応する仮想ブロック－物理ブロック変換と一致するように修正する手段を含む、請求項２７に記載のシステム。

【請求項３０】

前記コンピューティング環境が、コンピューティング・ユニットおよび外部記憶ユニッ

トを含み、前記外部記憶ユニットが、前記物理記憶装置を含み、前記使用する手段（ｉ）および前記使用する手段（ｉｉ）が、前記コンピューティング・ユニットによって実行される、請求項２３に記載のシステム。

【請求項３１】

クライアント・サーバ・コンピュータ環境でコピー・オン・ライトを容易にするシステムであって、

前記クライアント・サーバ・コンピューティング環境のファイルシステム・サーバで、ファイルに関する読取マッピング・テーブルおよび書込マッピング・テーブルを維持する手段

を含み、前記読取マッピング・テーブルが、修正のために物理記憶装置から前記ファイルのデータのブロックを読み取るのに使用される第１仮想ブロック－物理ブロック・マッピングを実行するのに使用可能であり、前記書込マッピング・テーブルが、前記ファイルの前記データの修正されたブロックを物理記憶装置に書き込むのに使用される第２仮想ブロック－物理ブロック・マッピングを実行するのに使用可能であり、前記読取マッピング・テーブルおよび前記書込マッピング・テーブルを使用して、データのブロックのコピー・オン・ライトが単一の書込動作を使用して達成可能である

システム。

【請求項３２】

前記ファイルに関する前記書込マッピング・テーブルが、前記読取マッピング・テーブルの対応する仮想ブロック－物理ブロック変換と異なる、前記物理記憶装置の物理ブロックにマッピングされる少なくとも１つの仮想ブロックを含む、請求項３１に記載のシステム。

【請求項３３】

前記ファイルのデータのブロックのコピー・オン・ライトが実行された後に前記読取マッピング・テーブルを更新する手段をさらに含み、前記更新する手段が、前記読取マッピング・テーブルの少なくとも１つの仮想ブロック－物理ブロック変換を、前記書込マッピング・テーブルの対応する仮想ブロック－物理ブロック変換と一致するように修正する手段を含む、請求項３１に記載のシステム。

【請求項３４】

複数のクライアントを有するクライアント・サーバ環境内でファイルのコピー・オン・ライトを実施するシステムであって、

（ｉ）前記クライアント・サーバ環境の第１クライアントで、コピー・オン・ライトされる前記ファイルのデータの少なくとも１つのブロックのコピー・オン・ライトを実行する手段と、

（ｉｉ）前記クライアント・サーバ環境の第２クライアントで、コピー・オン・ライトされる前記ファイルのデータの少なくとも１つの他のブロックのコピー・オン・ライトを実行する手段であって、前記ファイルの前記コピー・オン・ライトの異なる部分が、前記クライアント・サーバ環境内の前記複数のクライアントの異なるクライアントによって実行される、手段と

を含むシステム。

【請求項３５】

前記実行する手段（ｉ）が、前記第１クライアントによって、単一の書込動作を使用して前記ファイルのデータの前記少なくとも１つのブロックのコピー・オン・ライトを実行する手段を含み、前記実行する手段（ｉｉ）が、前記第２クライアントによって、単一の書込動作を使用してデータの前記少なくとも１つの他のブロックのコピー・オン・ライトを実行する手段を含む、請求項３４に記載のシステム。

【請求項３６】

前記クライアント・サーバ環境のファイルシステム・サーバが、前記ファイルを含む少なくとも１つの共用記憶装置に関連し、前記ファイルシステム・サーバが、前記少なくとも１つの共用記憶装置に保管された前記ファイルに関するマッピング・テーブルを維持し

、前記実行する手段（ｉ）が、前記第１クライアントによって、前記ファイルシステム・サーバから前記ファイルに関する読取マッピング・テーブルおよび書込マッピング・テーブルを入手し、前記ファイルのデータの前記少なくとも１つのブロックの前記コピー・アンド・ライトを実行する際に前記読取マッピング・テーブルおよび前記書込マッピング・テーブルを使用する手段を含み、前記実行する手段（ｉｉ）が、前記第２クライアントによって、前記ファイルシステム・サーバから前記ファイルに関する前記読取マッピング・テーブルおよび前記書込マッピング・テーブルを入手し、前記ファイルのデータの前記少なくとも１つの他のブロックの前記コピー・アンド・ライトを実行する際に前記読取マッピング・テーブルおよび前記書込マッピング・テーブルを使用する手段を含む、請求項３５に記載のシステム。

【請求項３７】

前記第１クライアントによって、前記第１クライアントが前記ファイルのデータの前記少なくとも１つのブロックのコピー・オン・ライトを実行したことを前記ファイルシステム・サーバに知らせ、それに応答して、前記ファイルシステム・サーバによって維持される前記ファイルに関する前記読取マッピング・テーブルおよび前記書込マッピング・テーブルの少なくとも１つを更新する手段をさらに含む、請求項３６に記載のシステム。

【請求項３８】

前記ファイルシステム・サーバが、前記第１クライアントによってコピー・オン・ライト更新された前記ファイルのデータの前記少なくとも１つのブロックおよび前記第２クライアントによってコピー・オン・ライト更新された前記ファイルのデータの前記少なくとも１つの他のブロックに対するコピー・オン・ライトを前記複数のクライアントのいずれもが行えなくする、請求項３６に記載のシステム。

【請求項３９】

前記クライアント・サーバ環境のファイルシステム・サーバが、前記ファイルを含む少なくとも１つの共用記憶装置に関連し、前記システムが、さらに、前記ファイルの前記コピー・オン・ライトを開始するために前記ファイルシステム・サーバによって前記ファイルを制御する手段を含み、前記開始する手段が、前記実行する手段（ｉ）および前記実行する手段（ｉｉ）によって使用される前記ファイルに関する書込マッピング・テーブルを更新する手段を含む、請求項３４に記載のシステム。

【請求項４０】

クライアント・サーバ環境内でファイルのコピー・オン・ライトを容易にするシステムであって、

前記クライアント・サーバ環境の共用記憶ユニットに保管されたファイルのコピー・オン・ライトの実施をファイルシステム・サーバから制御する手段であって、前記制御する手段が、前記クライアント・サーバ環境の第１クライアントが前記ファイル内のデータの部分をコピー・オン・ライトできるようにし、前記クライアント・サーバ環境の第２クライアントが前記ファイル内の前記データの異なる部分をコピー・オン・ライトできるようにする手段を含み、前記ファイルシステム・サーバが、前記ファイルの分散コピー・オン・ライトの実行を制御し、容易にする、制御する手段

を含むシステム。

【請求項４１】

前記制御する手段が、前記ファイルのコピー・オン・ライトを開始するために前記ファイルシステム・サーバによって前記ファイルを制御する手段を含み、前記開始することが、前記コピー・オン・ライトを実行するのに使用される前記ファイルに関する少なくとも１つのマッピング・テーブルを更新することを含む、請求項４０に記載のシステム。

【請求項４２】

前記ファイルシステム・サーバで、前記ファイルに関する読取マッピング・テーブルおよび書込マッピング・テーブルを維持する手段をさらに含み、前記読取マッピング・テーブルおよび前記書込マッピング・テーブルが、前記コピー・オン・ライトの実行に使用される、請求項４０に記載のシステム。

【請求項４３】
前記コピー・オン・ライトの実行の後に、前記ファイルに関する前記読取マッピング・テーブルおよび前記書込マッピング・テーブルの少なくとも１つを更新する手段をさらに含む、請求項４２に記載のシステム。
【請求項４４】
前記制御する手段が、前記ファイルシステム・サーバによって、前記ファイルの前記コピー・オン・ライトの一部としての、前記第１クライアントによってコピー・オン・ライト更新された前記ファイル内の前記データの前記部分に対する追加更新および前記第２クライアントによってコピー・オン・ライト更新された前記ファイル内の前記データの前記異なる部分に対する追加更新を防ぐ手段をさらに含む、請求項４０に記載のシステム。
【請求項４５】
製造品であって、
コンピューティング環境でコピー・オン・ライトを実施するコンピュータ可読プログラム・コード論理を有する少なくとも１つのコンピュータ使用可能媒体
を含み、前記コンピュータ可読プログラム・コード論理が、
（ｉ）ファイルのデータのブロックを修正のために物理記憶装置から読み取るのに使用される第１仮想ブロック－物理ブロック・マッピングを実行するのに第１マッピング・テーブルを使用する論理と、
（ｉｉ）前記ファイルの前記データの修正されたブロックを物理記憶装置に書き込むのに使用される第２仮想ブロック－物理ブロック・マッピングを実行するのに第２マッピング・テーブルを使用する論理であって、データの前記ブロックのコピー・オン・ライトが、単一の書込動作を使用して達成される、使用する論理と
を含む、製造品。
【請求項４６】
前記コンピューティング環境が、ファイルシステム・サーバおよび少なくとも１つのクライアントを含むクライアント・サーバ環境を含み、前記使用する論理（ｉ）および前記使用する論理（ｉｉ）が、前記クライアント・サーバ環境の前記少なくとも１つのクライアントによって実行される、請求項４５に記載の製造品。
【請求項４７】
前記少なくとも１つのクライアントによって実行される前記使用する論理（ｉ）および前記使用する論理（ｉｉ）が、前記ファイルのデータの前記ブロックのコピー・オン・ライトを実行する時に、前記第１マッピング・テーブルおよび前記第２マッピング・テーブルの少なくとも１つを入手するために前記ファイルシステム・サーバへの少なくとも１つの呼出しを行うことをさらに含む、請求項４５に記載の製造品。
【請求項４８】
前記コンピューティング環境が、コンピューティング・ユニットおよび外部記憶ユニットを含み、前記外部記憶ユニットが、前記物理記憶装置を含み、前記使用する論理（ｉ）および前記使用する論理（ｉｉ）が、前記コンピューティング・ユニットによって実行される、請求項４５に記載の製造品。
【請求項４９】
製造品であって、
クライアント・サーバ・コンピューティング環境でコピー・オン・ライトを容易にするコンピュータ可読プログラム・コード論理を有する少なくとも１つのコンピュータ使用可能媒体
を含み、前記コンピュータ可読プログラム・コード論理が、
前記クライアント・サーバ・コンピューティング環境のファイルシステム・サーバで、ファイルに関する読取マッピング・テーブルおよび書込マッピング・テーブルを維持する論理であって、前記読取マッピング・テーブルが、修正のために物理記憶装置から前記ファイルのデータのブロックを読み取るのに使用される第１仮想ブロック－物理ブロック・マッピングを実行するのに使用可能であり、前記書込マッピング・テーブルが、前記ファ

イルの前記データの修正されたブロックを物理記憶装置に書き込むのに使用される第２仮想ブロック－物理ブロック・マッピングを実行するのに使用可能であり、前記読取マッピング・テーブルおよび前記書込マッピング・テーブルを使用して、データのブロックのコピー・オン・ライトが単一の書込動作を使用して達成可能である、論理
を含む、製造品。
【請求項５０】
製造品であって、
複数のクライアントを有するクライアント・サーバ環境内でファイルのコピー・オン・ライトを実施するコンピュータ可読プログラム・コード手段を有する少なくとも１つのコンピュータ使用可能媒体
を含み、前記コンピュータ可読プログラム・コード手段が、
前記クライアント・サーバ環境の複数のクライアントを使用してファイルのコピー・オン・ライトを実行する論理
を含み、前記実行する論理が、
（ｉ）前記複数のクライアントの第１クライアントによって、前記ファイルのデータの少なくとも１つのブロックのコピー・オン・ライトを実行する論理と、
（ｉｉ）前記複数のクライアントの第２クライアントによって、前記ファイルのデータの少なくとも１つの他のブロックのコピー・オン・ライトを実行する論理と
を含む、製造品。
【請求項５１】
前記論理（ｉ）が、前記第１クライアントによって、単一の書込動作を使用して前記ファイルのデータの前記少なくとも１つのブロックのコピー・オン・ライトを実行する論理を含み、前記論理（ｉｉ）が、前記第２クライアントによって、単一の書込動作を使用してデータの前記少なくとも１つの他のブロックのコピー・オン・ライトを実行する論理を含む、請求項５０に記載の製造品。
【請求項５２】
前記クライアント・サーバ環境のファイルシステム・サーバが、前記ファイルを含む少なくとも１つの共用記憶装置に関連し、前記ファイルシステム・サーバが、前記少なくとも１つの共用記憶装置に保管された前記ファイルに関するマッピング・テーブルを維持し、前記論理（ｉ）が、前記第１クライアントによって、前記ファイルシステム・サーバから前記ファイルに関する読取マッピング・テーブルおよび書込マッピング・テーブルを入手し、前記ファイルのデータの前記少なくとも１つのブロックの前記コピー・アンド・ライトを実行する際に前記読取マッピング・テーブルおよび前記書込マッピング・テーブルを使用する論理を含み、前記論理（ｉｉ）が、前記第２クライアントによって、前記ファイルシステム・サーバから前記ファイルに関する前記読取マッピング・テーブルおよび前記書込マッピング・テーブルを入手し、前記ファイルのデータの前記少なくとも１つの他のブロックの前記コピー・アンド・ライトを実行する際に前記読取マッピング・テーブルおよび前記書込マッピング・テーブルを使用する論理を含む、請求項５１に記載の製造品。
【請求項５３】
製造品であって、
クライアント・サーバ環境内でファイルのコピー・オン・ライトを容易にするコンピュータ可読プログラム・コード論理を有する少なくとも１つのコンピュータ使用可能媒体
を含み、前記コンピュータ可読プログラム・コード論理が、
前記クライアント・サーバ環境の共用記憶ユニットに保管されたファイルのコピー・オン・ライトの実施をファイルシステム・サーバから制御する論理であって、前記制御する論理が、前記クライアント・サーバ環境の第１クライアントが前記ファイル内のデータの部分をコピー・オン・ライトできるようにする論理と、前記クライアント・サーバ環境の第２クライアントが前記ファイル内の前記データの異なる部分をコピー・オン・ライトできるようにする論理とを含み、前記ファイルシステム・サーバが、前記ファイルの分散コ

ピー・オン・ライトの実行を制御し、容易にする、制御する論理
を含む、製造品。
【請求項５４】
前記制御する論理が、前記ファイルのコピー・オン・ライトを開始するために前記ファイルシステム・サーバによって前記ファイルを制御する論理を含み、前記開始することが、前記コピー・オン・ライトを実行するのに使用される前記ファイルに関する少なくとも１つのマッピング・テーブルを更新することを含む、請求項５３に記載の製造品。
【請求項５５】
前記ファイルシステム・サーバで、前記ファイルに関する読取マッピング・テーブルおよび書込マッピング・テーブルを維持する論理をさらに含み、前記読取マッピング・テーブルおよび前記書込マッピング・テーブルが、前記コピー・オン・ライトの実行に使用される、請求項５３に記載の方法。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、全般的にはコンピューティング環境内のファイルシステム・データ管理に関し、具体的には、さまざまなコンピューティング環境内のファイルシステム・データ・ファイルのコピー・オン・ライト（copy-on-write）を実施する技法に関する。
【背景技術】
【０００２】
汎用コンピュータおよびデータ処理システムを含む多数のタイプのコンピューティング環境で、「仮想記憶」方式を使用して編成された記憶装置が使用される。一般的な仮想記憶を用いると、コンピューティング環境で実行中のアプリケーションまたはプロセスあるいはその両方が、自由に使える無制限の量のメモリを有するかのように振る舞えるようになる。実際には、特定のアプリケーションまたはプロセスから使用可能な記憶装置の量は、コンピューティング環境内の記憶装置の量によって制限され、さらに、その記憶装置を共用する並行に実行中のプログラムの数によって制限される。さらに、仮想記憶方式によって、メモリの実際の物理アドレスがアプリケーション・プログラムから隠蔽される。アプリケーション・プログラムは、論理アドレスを使用してそのプログラムのメモリ空間にアクセスし、この論理アドレスが、コンピューティング環境によって物理アドレスに変換される。
【０００３】
仮想記憶システムによって、記憶装置が「ブロック」（または「ページ」）と称する単位で編成される。これらのブロックは、高速な主記憶と、１つまたは複数のより大きいが通常は低速の二次、三次などの記憶ユニットの間で移動される。ブロックの移動（しばしばスワップと称する）は、コンピューティング環境で実行されるアプリケーションまたはプロセスには透過的であり、アプリケーションまたはプロセスは、それぞれが無制限の量のメモリを有するかのように振る舞うことができる。
【０００４】
ある種の従来のシステムは、時々、メモリの諸部分をコピーすることを必要とする。このコピーは、ユーザが開始するか、オペレーティング・システムによって開始されるのいずれかとすることができる。従来のシステムでは、しばしば、「フラッシュ・コピー（flash
copy）」の「遅延（lazy）」コピー方法が使用され、この場合に、コピーされる記憶装置が、読取専用の状況を割り当てられるが、実際のコピーは後に延期される。オリジナルまたはコピーのいずれかへの書込の試みが行われる場合に、メモリが、その時にコピーされ、オリジナルとコピーの両方に、読取－書込の入出力状況が与えられる。この形で、コピーが即座に行われるように見えるが、実際のコピーは、可能な最後の時まで延期される。書込が実行されない場合には、コピーは行われない。この理由から、この方法が、「コピー・オン・ライト」または「仮想コピー（virtualcopy）」と呼ばれる。

【０００５】
一般に、コピー・オン・ライト動作は、単一の書込が２つの書込動作をもたらすので、計算的に高価である。すなわち、既存のデータ・ブロックを、古い物理ブロックから新しい物理ブロックにコピーする必要があり、その後、実際の更新／書込動作が、新しい物理ブロックに対して実行される。
【非特許文献１】ランダル・クライトン・バーンズ（Randal ChiltonBurns）著、「ストレージ・エリア・ネットワーク用の分散ファイル・システムでのデータ管理（DataManagement In A DistributedFile System For StorageArea Networks）」、カリフォルニア大学（University of California）、米国サンタクルーズ（SantaCruz）、２０００年３月
【発明の開示】
【発明が解決しようとする課題】
【０００６】
この計算オーバヘッドに鑑みて、当技術分野に、二重書込要件を部分的に除去する新規のコピー・オン・ライト実施形態の必要がある。
【課題を解決するための手段】
【０００７】
一態様では、コンピューティング環境でコピー・オン・ライトを実施する方法を介して、従来技術の短所が克服され、追加の利益がもたらされる。この方法には、ファイルのデータのブロックを修正のために物理記憶装置から読み取るのに使用される第１仮想ブロック－物理ブロック・マッピングを実行するのに第１マッピング・テーブルを使用することと、前記ファイルの前記データの修正されたブロックを物理記憶装置に書き込むのに使用される第２仮想ブロック－物理ブロック・マッピングを実行するのに第２マッピング・テーブルを使用することであって、データの前記ブロックのコピー・オン・ライトが、単一の書込動作を使用して達成される、使用することとが含まれる。
【０００８】
もう１つの態様では、クライアント・サーバ・コンピューティング環境でコピー・オン・ライトを容易にする方法が提供される。この方法には、前記クライアント・サーバ・コンピューティング環境のファイルシステム・サーバで、ファイルに関する読取マッピング・テーブルおよび書込マッピング・テーブルを維持することが含まれ、前記読取マッピング・テーブルが、修正のために物理記憶装置から前記ファイルのデータのブロックを読み取るのに使用される第１仮想ブロック－物理ブロック・マッピングを実行するのに使用可能であり、前記書込マッピング・テーブルが、前記ファイルの前記データの修正されたブロックを物理記憶装置に書き込むのに使用される第２仮想ブロック－物理ブロック・マッピングを実行するのに使用可能であり、前記読取マッピング・テーブルおよび前記書込マッピング・テーブルを使用して、データのブロックのコピー・オン・ライトが単一の書込動作を使用して達成可能である。
【０００９】
もう１つの態様では、複数のクライアントを有するクライアント・サーバ環境内でファイルのコピー・オン・ライトを実施する方法が提供される。この方法には、前記クライアント・サーバ環境の複数のクライアントを使用してファイルのコピー・オン・ライトを実行することが含まれる。実行することに、前記複数のクライアントの第１クライアントによって、コピー・オン・ライトされる前記ファイルのデータの少なくとも１つのブロックのコピー・オン・ライトを実行することと、前記複数のクライアントの第２クライアントによって、コピー・オン・ライトされる前記ファイルのデータの少なくとも１つの他のブロックのコピー・オン・ライトを実行することとが含まれる。機能強化された態様では、実行することに、前記第１クライアントによって、単一の書込動作を使用して前記ファイルのデータの前記少なくとも１つのブロックのコピー・オン・ライトを実行することと、前記第２クライアントによって、やはり単一の書込動作を使用してデータの前記少なくとも１つの他のブロックのコピー・オン・ライトを実行することとが含まれる。
【００１０】

もう１つの態様では、クライアント・サーバ環境内でファイルのコピー・オン・ライトを容易にする方法が提示される。この方法には、前記クライアント・サーバ環境の共用記憶ユニットに保管されたファイルのコピー・オン・ライトの実施をファイルシステム・サーバから制御することであって、前記制御することが、前記クライアント・サーバ環境の第１クライアントが前記ファイル内のデータの部分をコピー・オン・ライトできるようにすることと、前記クライアント・サーバ環境の第２クライアントが前記ファイル内の前記データの異なる部分をコピー・オン・ライトできるようにすることとを含み、前記ファイルシステム・サーバが、前記ファイルの分散コピー・オン・ライトの実行を制御し、容易にする、制御することが含まれる。

【００１１】

上で要約した方法のさまざまな追加の特徴および機能強化も、上で要約した方法に対応するシステムおよびコンピュータ・プログラム製品と同様に、本明細書に記載され、請求される。

【００１２】

さらに、本発明の教示を介して、追加の特徴および長所が実現される。本発明の他の実施形態および態様は、本明細書で詳細に説明され、請求される発明の一部とみなされる。

【００１３】

発明とみなされる主題は、本明細書の先頭の請求項で具体的に指摘され、明確に請求される。本発明の前述および他の目的、特徴、および長所は、添付図面と共に解釈される下記の詳細な説明から明白になる。

【発明を実施するための最良の形態】

【００１４】

概要

本明細書で提示されるのは、一態様で、コンピューティング環境でコピー・オン・ライトを実施する技法である。この技法には、異なる変換すなわち、読取マッピング・テーブルおよび書込マッピング・テーブルを使用して、単一書込動作を使用してファイル内のデータの単位のコピー・オン・ライトを達成することが含まれる。一例として、コピー・オン・ライトは、第１の仮想ブロック－物理ブロック・マッピングを使用して、修正のために物理記憶装置からファイルのデータのブロックを読み取ることと、その後、第２の仮想ブロック－物理ブロック・マッピングを使用してそのデータの修正されたブロックを物理記憶装置に書き込むことによって達成され、第１仮想ブロック－物理ブロック・マッピングおよび第２仮想ブロック－物理ブロック・マッピングに、異なるマッピングが含まれる。

【００１５】

もう１つの態様で、本明細書に提示されるのは、クライアント・サーバ環境の複数のクライアントにまたがるファイルの分散コピー・オン・ライトを実施する技法である。そのような環境内で、複数のクライアントの第１クライアントによって、ファイルのデータの少なくとも１ブロックのコピー・オン・ライトが実行され、複数のクライアントの第２クライアントによって、ファイルのデータの少なくとも１つの他のブロックのコピー・オン・ライトが実行される。一実施形態では、クライアントに、異種オペレーティング・システムを含めることができ、コピー・オン・ライトされるファイル内のデータのブロックのコピー・オン・ライトのそれぞれが、単一の書込動作を使用して実行される。また、コピー・オン・ライトを、上で要約したように第１マッピング変換（たとえば、読取マッピング・テーブルを使用する）および第２マッピング変換（たとえば、書込マッピング・テーブルを使用する）を使用して達成することができる。本発明の上記および他の態様を、下で説明し、請求項に列挙する。

【００１６】

詳細な説明

全体的に１００で示される、本発明の態様によるコピー・オン・ライトが組み込まれ、使用されるコンピューティング環境の例を、図１に示す。図からわかるように、コンピュ

ーティング環境１００には、たとえば、少なくとも１つの中央処理装置１０２、メモリ１０４、および１つまたは複数の記憶ユニットまたは記憶装置１０６が含まれる。

【００１７】

既知のように、中央処理装置１０２は、コンピューティング・ユニットの制御センタであり、これによって、命令実行、割込みアクション、タイミング機能、初期プログラム・ロード、および他の機械関連機能のシーケンシング機能および処理機能が提供される。中央処理装置によって、少なくとも１つのオペレーティング・システムが実行され、オペレーティング・システムは、既知のように、他のプログラムの実行を制御し、周辺装置との通信を制御し、コンピュータ・リソースの使用を制御することによって、コンピューティング・ユニットの動作を制御するのに使用される。

【００１８】

中央処理装置（ＣＰＵ）１０２は、メモリ１０４に結合され、このメモリ１０４は、直接にアドレス可能であり、メモリ１０４によって、中央処理装置によるデータの高速処理がもたらされる。メモリ１０４には、下でさらに説明するようにＣＰＵ１０２によって使用されるバッファまたはキャッシュ領域１０３が含まれる。もう１つの実施形態では、バッファ１０３を、ＣＰＵ１０２内に置くことができる。記憶ユニット１０６は、入出力装置の一例である。本明細書で使用される記憶ユニット１０６は、コンピューティング・ユニットの外部またはコンピューティング環境１００のコンピューティング・ユニット内とすることができ、記憶ユニット１０６には、たとえば、主記憶、磁気記憶媒体（たとえば、テープ、ディスク）、および直接アクセス記憶装置などを含めることができる。データを、図示のように、ＣＰＵ１０２、メモリ１０４、および記憶ユニット１０６の間で転送することができる。

【００１９】

一例では、コンピューティング環境１００が、単一のシステム環境であり、ＡＩＸオペレーティング・システムで動作するＲＳ／６０００コンピュータ・システムが含まれる（ＲＳ／６０００およびＡＩＸは、International Business Machines Corporation社によって提供される）。しかし、本発明は、そのような環境に制限されない。本発明の機能を、多数の他のタイプのコンピュータ環境および多数のタイプのコンピュータ・システムに組み込み、使用することができる。たとえば、コンピュータ環境１００に、分散コンピューティング環境を含めることができ、ＵＮＩＸ（Ｒ）ベースのオペレーティング・システムが動作するＵＮＩＸ（Ｒ）ワークステーションを含めることができる。他の変形形態も、可能であり、請求される発明の一部とみなされる。

【００２０】

既知のように、ファイルは、ユーザ／アプリケーション・データの保管に使用することができる、コンピューティング環境のファイルシステム内の名前付きオブジェクトである。このデータには、ファイル名、オフセット、および長さを指定することによってアクセスすることができる。ユーザ・アプリケーションまたはプロセスにとって、ファイル上のデータは、連続的に見えるが、記憶ユニット（ディスクなど）内では、データ表現が異なる可能性がある。各ファイルシステムによって、仮想（相対）オフセット・ブロック番号から物理ブロック番号への間のマッピングまたは変換を提供するマッピング・テーブルが維持され、ブロックは、ファイル内のデータのページまたは他の単位とすることができ、この単位のサイズは、ファイルシステムによって指定される。

【００２１】

図１の例では、ファイルシステムが、記憶ユニットを含むと仮定され、記憶ユニットは、やはり、コンピューティング環境の特定のコンピューティング・ユニットに外付けまたは内蔵とすることができる。図２に、アプリケーションまたはプロセスのデータ範囲およびオフセットを、所与のファイルの仮想／相対ブロック番号に相関させるファイル表現２００の一例を示す。各ファイルが、それ自体の仮想／相対ブロック番号の組を有することに留意されたい。この例では、０から４Ｋバイトまでのファイル・データが、仮想／相対

ブロック番号１にマッピングされ、４Ｋから８Ｋバイトまでのデータが、ブロック番号２にマッピングされ、８Ｋから１２Ｋバイトまでののデータが、ブロック番号３にマッピングされ、１２Ｋから１６Ｋバイトまでのデータが、ブロック番号４にマッピングされる。やはり、これらの数は、例としてのみ提供される。

【００２２】

図３に、特定のファイルのファイルシステム・マッピング・テーブル３００の一例を示す。テーブル３００は、ファイルの仮想／相対ブロック番号を記憶ユニットの実際の物理ブロック・アドレスに変換する際に使用される。たとえば、仮想／相対ブロック番号１、２、３、および４が、記憶ユニット４００の物理ブロック・アドレスＡ、Ｄ、Ｇ、およびＬにマッピングされることが示されている（図４参照）。

【００２３】

たとえば、アプリケーションまたはプロセスによって、特定のファイルのオフセット５０００からの４バイトのデータの読取が望まれる場合に、図２および図３のファイル表現２００およびファイルシステム・マッピング・テーブル３００を使用すると、実際のデータ読取は、図５の物理ブロックＤから行われることがわかる。これは、オフセット５０００から始まる４バイトのデータが、仮想／相対ブロック番号２に含まれ、これが、図３に示されているように、記憶ユニットの物理ブロック・アドレスＤに変換されるからである。

【００２４】

最初に注記したように、フラッシュ・コピー動作によって、記憶装置の空間効率のよいコピーをすばやく行えるようになる。この動作は、高速である必要があるので、物理的なコピーは、当初は、動作の一部として作られない。後に、適用可能なファイル・データを修正する試みによって、コピー・オン・ライト動作がもたらされる。クライアント・サーバ環境では、メタデータ・コピー・オン・ライトが、通常は、ファイルシステム・サーバによって実行され、ファイル・データ・コピー・オン・ライトは、クライアントによって実行される。PageInスレッドおよびPageOutスレッドを使用して、データのブロックをクライアントのキャッシュに持ち込み、そのデータを更新し、その後、そのデータを記憶ユニットに書き戻すことができる。本明細書で説明するように、PageInおよびPageOutについて異なる変換が使用される場合に、クライアントは、潜在的なコピー・オン・ライト・データをバッファに読み取り、バッファ内のデータに更新を適用し、修正されたデータを、PageOutスレッドを介して記憶ユニットの新しい位置に書き込むことができる。したがって、２つのマッピング・テーブルまたは変換を有することによって、一実施形態で、既存のPageInおよびPageOutの概念を活用するコピー・オン・ライト技法がもたらされる。

【００２５】

図６は、書込動作ならびに本発明の態様によるコピー・オン・ライト動作の一実施形態の流れ図実施形態である。書込動作６００は、ファイルのデータのブロック全体（またはページ全体）を書き込むかどうかを判定すること６１０によって開始される。データの部分ブロック書込の場合には、ファイル内のデータの適用可能なブロックを、記憶ユニットからローカル・バッファに読み取る６２０（PageIn）。バッファ内のデータのブロックの更新６３０がこれに続き、その後、ファイルのデータの修正されたブロックの記憶ユニットへの書込６４０が続く。ファイルのデータのブロック全体を書き込む場合には、この論理では、単純に、記憶ユニット（図６の６５０）へのデータのブロック全体の書込に進む。既知のように、図６の論理は、オペレーティング・システム・カーネル内で実施することができる。図７および８に、図６の論理を使用して書込動作を実施するさらなる処理の例を示す。

【００２６】

図７に、ファイルのデータのブロックを記憶ユニットからバッファへ読み取る処理の一実施形態を示す。アプリケーション入力は、やはり、読み取られるファイル・データのオフセットおよび長さであり７００、これは、仮想ブロック番号を計算する７１０のに使用される（たとえば、図２に示されたものなどのファイル表現を使用して）。次に、サブル

ーチン呼出しFile・Getを実行して、仮想ブロック番号を、記憶ユニット内のファイル・データの物理ブロック・アドレスにマッピングする７２０。物理ブロック・アドレスを使用して、データのブロックを、記憶ユニットからローカル・バッファに読み取る７３０。

【００２７】

図８に、ファイルのデータのブロックを記憶ユニットに書き込む処理の一例を示す。図からわかるように、仮想ブロック番号８００を、File・Getマッピング処理で使用して、たとえば図３に示されたものなどのファイルシステム・マッピング・テーブルを使用して、仮想－物理マッピングを入手する８１０。この物理ブロック・アドレスが、修正されたデータ・ブロックをバッファから記憶ユニットに書き込む時８２０に使用される。一実施形態では、図７のブロック読取処理に、PageInスレッド処理を含めることができ、図８のデータ・ブロック書込プロセスに、PageOutスレッド処理を含めることができる。

【００２８】

有利なことに、本明細書で開示されるのは、図６の高水準書込論理フローの変更なしでコピー・オン・ライトを達成する技法である。この技法では、コピー・オン・ライトが実行されるデータの特定のファイルに関する、読取マッピング・テーブルおよび書込マッピング・テーブルと称する２組の変換またはマッピング・テーブルが使用される。一実施形態で、コピー・オン・ライトが実行される時に、必ず、これらの２つのマッピング・テーブルが、ファイルシステムによって維持され、クライアント・アプリケーションによってアクセスされる。たとえば、この２つのマッピング・テーブルが、物理アドレス変換境界でファイルシステム・ドライバに同時に提示される。

【００２９】

たとえば、図９に、読取マッピング・テーブル９００および書込マッピング・テーブル９１０を示す。読取マッピング・テーブル９００によって、仮想ブロック番号１、２、３、および４が、それぞれ物理ブロック番号Ａ、Ｄ、Ｇ、およびＬにマッピングされ、書込マッピング・テーブル９１０によって、仮想ブロック番号１、２、３、および４が、それぞれ物理ブロックＷ、Ｘ、Ｙ、およびＺにマッピングされる。読取マッピング・テーブルによって、読取動作に使用される第１の仮想－物理変換が提供され、書込マッピング・テーブルによって、書込動作に使用される第２の仮想－物理変換が提供される。具体的に言うと、一例として、コピー・オン・ライトを、PageInについて読取テーブル変換、PageOutについて書込テーブル変換を使用して実施することができる。

【００３０】

図１０に、図６の論理によるコピー・オン・ライト動作に使用されるデータ・ブロック読取処理の実施形態を示す。図からわかるように、アプリケーションによって、ファイル内のデータ・オフセットおよび長さが指定され１０００、これから仮想ブロック番号が計算される１０１０（図２参照）。次に、ファイルに関する読取マッピング・テーブル（たとえば図９のテーブル９００）を使用して、仮想－物理読取マッピングを入手する１０２０。次に、この物理読取マッピングを使用して、ファイルの少なくとも１つのデータ・ブロックを記憶ユニットからローカル・バッファに読み取る１０３０。

【００３１】

図１１に、図６の論理によるコピー・オン・ライト動作に関するデータ・ブロック書込処理の例を示す。仮想ブロック番号１１００を使用して、ファイルの書込マッピング・テーブル（たとえば図９のテーブル９１０）を使用して、仮想－物理「書込」マッピング１１１０を入手する。物理書込マッピングを使用して、修正されたデータ・ブロックを、ローカル・バッファから対応する記憶ユニット物理ブロックに書き込む１１２０。上で注記したように、ファイルのデータのブロック全体のコピー・オン・ライトが実行される場合に、修正されたデータ・ブロックが、図１１の物理「書込」マッピングを使用して、単純に記憶装置に直接に書き込まれる。その場合には、ファイルの書込マッピング・テーブルだけが使用される。

【００３２】

当業者は、読取、更新、および書込が記憶ユニット内の同一の物理ブロック・アドレス

で行われるようにするために、読取マッピング・テーブルおよび書込マッピング・テーブルを同一にすることによって、図６、１０、および１１の論理を使用して通常の書込動作を実行できることに気付くであろう。しかし、有利なことに、この論理は、コピー・オン・ライトについても同一である。コピー・オン・ライトが、ファイルについて要求される時に、ファイルシステムによって、たとえば新しい物理アドレス・ブロックが割り振られ、そのファイルの対応する書込マッピング・テーブルが更新される。それを行うことによって、書込動作が、異なる物理ブロック・アドレスで行われ、読取動作もそのアドレスで行われ、これは、オリジナルのデータが記憶ユニット内で手付かずのままになることを意味する。書込動作の後に、そのファイルの読取マッピング・テーブルを、特定のコピー・オン・ライト・アプリケーションに応じて更新することができる。

【００３３】

図９のマッピング・テーブルを参照し、図５の例を使用すると、コピー・オン・ライト変更が、ファイルのオフセット５０００の４バイトのデータに対して行われる（たとえば、データ内容「１　２」を「２　１」に変更する）場合に、データの更新されたブロックが、記憶ユニットの物理ブロック番号Ｘ（書込マッピング・テーブル９１０（図９）を使用すると、仮想ブロック番号２に対応する）に書き込まれる。コピー・オン・ライト動作の後に、対応する読取マッピング・テーブル（図９の９００）を、仮想ブロック番号２が記憶ユニット内の物理ブロック番号Ｘに変換されるように更新することができ、これを、図１２の更新された読取マッピング・テーブル１２００に示す。図１２の書込マッピング・テーブル１２１０は、図９の書込マッピング・テーブル９１０と同一である。しかし、コピー・オン・ライトの後に、読取マッピング・テーブル内の物理ブロック番号が更新されるかどうか、およびどのように更新されるかは、コピー・オン・ライトを使用するアプリケーションに依存する。たとえば、「データ保存」タイプのアプリケーションでは、読取マッピング・テーブル内の物理ブロック番号を、初期ファイル参照を保存するように維持することができる。ファイル・データが記憶ユニットのある部分から別の部分に物理的に移動される「データ移動」タイプのアプリケーションでは、書込マッピング・テーブルからの新しい物理ブロック・アドレス番号を用いて読取マッピング・テーブルを更新する際に、元の物理ブロック番号を解放することができる。

【００３４】

図１３に、全体的に１３００で示される、本発明の１つまたは複数の態様を組み込み、使用することができるコンピューティング環境のもう１つの例を示す。環境１３００には、クライアント１　１３１０およびクライアント２　１３２０を含む複数のクライアントが含まれ、これらのクライアントは、この例では、クライアント／サーバ通信に関するインターネット・プロトコル・ネットワークによってファイルシステム・サーバ１３３０に接続される。サーバ１３３０は、ストレージ・エリア・ネットワーク１３４０に接続され、ストレージ・エリア・ネットワーク１３４０は、ファイル・データの保管に使用可能な複数のストレージ・プール１３５０を有する。クライアント１　１３１０およびクライアント２　１３２０は、ストレージ・エリア・ネットワーク１３４０にも直接に接続される。

【００３５】

一例として、コンピューティング環境１３００は、１つの位置（たとえばファイルシステム・サーバ）での本明細書に記載のものなどのマッピング・ファイル（たとえば、ファイルの読取マッピング・テーブルおよび書込マッピング・テーブル）の維持、クライアント・アプリケーションがストレージ・エリア・ネットワーク（ＳＡＮ）を介して記憶ユニット（すなわち複数のストレージ・プール）に直接にアクセスできること、およびクライアント・アプリケーションが複数のストレージ・プール内のファイルの任意のオブジェクトへの読取／書込アクセスを有することを含む、ある種の特徴を有すると仮定される。そのようなＳＡＮ環境は、ランダル・クライトン・バーンズ（Randal Chilton Burns）著、「ストレージ・エリア・ネットワーク用の分散ファイル・システムでのデータ管理（DataManagement

In A Distributed File System For Storage Area Networks）」、カリフォルニア大学（Universityof California）、米国サンタクルーズ（Santa Cruz）、２０００年３月による理論を含めて、さまざまな刊行物に詳細に記載されている。

【００３６】

本発明のもう１つの態様では、異なるクライアント・アプリケーションが１つのファイルの異なる部分を更新する、分散コピー・オン・ライト機能が提示される。たとえば、クライアント１が、ファイルの仮想ブロック２のコピー・オン・ライトを実行し、その間に、クライアント２が、そのファイルの仮想ブロック４のコピー・オン・ライトを実行する。図１４および１５に、この例を詳細に示す。

【００３７】

図１４に、仮想ブロック２のコピー・オン・ライトを実行するクライアント１の一例を示す。クライアント１は、まず、ファイルシステム・サーバから、「my file」というラベルを付けられたファイルのマッピング・テーブルを入手する１４００。たとえばオフセット５０００で更新する間に、コピー・オン・ライトが、仮想ブロック２に対して実行される１４１０。クライアント１は、仮想ブロック２のコピー・オン・ライトを実行したことをファイルシステム・サーバに知らせ１４２０、ファイルシステム・サーバが、それ相応にマッピング・テーブルを更新する１４３０。一例では、コピー・オン・ライトを、図６および９から１２に関して上で説明したように実行することができる。

【００３８】

クライアント１は、ロック機構を使用して、ファイル・データの仮想ブロック２のコピー・オン・ライト更新を実行する。ロック機構には、ファイルシステム・オブジェクトごとのロックが含まれ、クライアントは、所与のファイルに対して動作を実行するために、ファイルシステム・サーバからそのロックを獲得する必要がある。したがって、一実施形態では、クライアント１は、このロックを受け取る時に、そのファイルのマッピング・テーブルも受け取り、クライアント１がロックを失う時に、そのファイルに関するクライアント１側のすべてのマッピング・テーブルが無効になる。したがって、クライアント１が次にロックを入手する時に、クライアント１は、既存のマッピング・テーブルを使用することができず、その代わりに、ファイルシステム・サーバから現在のマッピング・テーブルを入手する。

【００３９】

さらなる説明として、図１３のコンピューティング環境で、分散ロック機構を使用して、異なるクライアントからのファイルシステム・オブジェクトへのアクセスを制御することができる。分散ロックには２つのタイプすなわちセッション・ロックとデータ・ロックがある。これらのロックは、ファイルシステム・オブジェクトごとのロックである。セッション・ロックは、オープン「ファイル記述子」と同等である。このロックがクライアントによって獲得された時に、このクライアントがこのファイルを使用することに関心を持っていることがサーバに知らされる。クライアントは、異なるモード、たとえば読取モード、書込モード、排他モードなどでセッション・ロックを獲得することができる。たとえば、クライアントＡが、排他モードでセッション・ロックを保持している時に、もう１つのクライアント、クライアントＢが、同一のファイルのオープンを望む時に、クライアントＢは、セッション・ロックに関する要求をサーバに送り、サーバは、クライアントＡが排他モードのロックを有するのでそのロック要求を拒否し、したがって、クライアントＢは、ファイルを操作することができない。第２のタイプの分散ロックが、データ・ロックである。データ・ロックは、物理的な読取／書込を行うのに使用される。クライアントは、ファイルに対する読取／書込を行うために、読取／書込モードのデータ・ロックを有しなければならない。さらなる説明として、セッション・ロックを、２つのクライアントが書込モードで同時に保持することができる。これは、両方のクライアントが、特定のファイルに書き込むアクセス権を有することを意味するが、これらのクライアントは、書込モードでデータ・ロックを入手するまでは書き込むことができない。しかし、所与の時点で、１つのクライアントだけが、「書込モード」のデータ・ロックを得ることができ、その

結果、１つのクライアントだけが、実際の入出力を実行することができる。両方のクライアントが、能動的にファイルに作用している場合に、これらのクライアントは、書込モードのセッション・ロックを保持しているが、データ・ロックは、書込を完了するために両方のクライアントの間を往復する。

【００４０】

ある時点で、クライアント２が、ファイル「my file」の別の部分のコピー・オン・ライトを実行するためにロックを要求する。図１５からわかるように、クライアント２は、ファイル「myfile」のマッピング・テーブルをファイルシステム・サーバから入手する１５００。オフセット１４０００（すなわち仮想ブロック番号４）でファイル内容を更新する間に、クライアント２は、ファイルの仮想ブロック番号４に対するコピー・オン・ライトを実行する１５１０。その後、クライアント２は、このコピー・オン・ライト更新に関してファイルシステム・サーバに知らせ、サーバは、それ相応に「myfile」の読取マッピング・テーブルを更新する１５３０。図１６に、図１４および１５の「my file」のコピー・オン・ライト更新から生じる更新された読取マッピング・テーブル１６００および書込マッピング・テーブル１６１０（図９のマッピング・テーブルから開始された）を示す。図からわかるように、読取マッピング・テーブルは、仮想ブロック番号２を物理ブロック番号Ｘにマッピングされ、仮想ブロック番号４を物理ブロック番号Ｚにマッピングされている。

【００４１】

要約すると、図１３に示されたものなどの環境内でのコピー・オン・ライトの実施形態を、複数のクライアント・アプリケーションにまたがって分散させることができる。したがって、複数のクライアント・アプリケーションを、分散ファイルシステムでのファイルのコピー・オン・ライトの実行に用いることができる。上で説明したものなど、１時に１つのクライアント・アプリケーションだけによって所有される、分散ロック機構を使用することができる。たとえば、クライアント１が、分散ロックを有する時に、クライアント１は、コピー・オン・ライト動作に関する有効な読取マッピング・テーブルおよび書込マッピング・テーブルを有する。クライアント２が、同一のファイルの別の部分のコピー・オン・ライトを行うことを望む場合に、ファイルシステム・サーバは、クライアント１から分散ロックを奪い、クライアント２に与える。代替の例では、分散ロックを、ファイル範囲によって分割することができ、その結果、クライアント１およびクライアント２の両方が、ファイルの異なる部分に対するコピー・オン・ライトを同時に実行できるようになる。

【００４２】

図１７に、分散クライアント・サーバ環境でのコピー・オン・ライトに関する「データ保存」アプリケーションの一例を示す。まず、管理者は、ファイル「my file」のポイント・イン・タイム・イメージを作ると決定する１７００。管理コマンドを実行して、「my file」を保存し１７１０、次に、ファイルシステム・サーバが、クライアント・アプリケーションからファイル「myfile」の制御を奪う１７２０。たとえば、ファイルシステム・サーバは、そのファイルに関するすべての分散ロックをクライアント・アプリケーションから奪う（このステップは、非分散非クライアント／サーバ環境では適用可能でない）。次に、ファイルシステム・サーバが、書込マッピング・テーブルを更新してコピー・オン・ライトを促進し１７３０、サーバが、めいめいのクライアント・アプリケーションに制御を返す１７４０。この点から後は、クライアント・アプリケーションによる「myfile」に対するすべての更新が、ファイルのその特定の部分に関してコピー・オン・ライトが実行されることを自動的にもたらすと同時に、古いデータ・パスが保存される（すなわち、読取マッピング・テーブルの物理ブロック番号を、初期ファイル参照を保存するために維持することができる）。

【００４３】

特定の例

本発明のさまざまな態様の１つの詳細な実施形態を、次に提示する。この詳細な説明で

は、２タイプの入出力動作すなわち、バッファ入出力および直接入出力が可能である。
【００４４】
バッファ付き入出力
バッファ付き入出力は、バッファ・キャッシュを介して実行される入出力を意味する。この場合に、読取／書込は、まずキャッシュに頼り、その後、このキャッシュデータが、記憶ユニット（たとえばディスク）にハードニング（harden）される。
【００４５】
既存ファイルに対する更新は、データをキャッシュに読み取ることによって行われる。すべての変更をキャッシング／バッファリングされたページに適用することと、そのデータをディスクに書き戻すこと。この事実を考慮に入れると、本明細書で開示されるコピー・オン・ライトは、追加コストなしで達成される。たとえば、分散ファイルシステムにアクセスするクライアントに、少なくとも２つの構成要素すなわち、（ｉ）オペレーティング・システム固有であり、アプリケーションからの要求を処理し、バッファ・メモリ・システムおよびディスクなどの記憶装置と通信するインストール可能ファイル・システム（ＩＦＳ）、および（ｉｉ）すべてのオペレーティング・システムに共通し、ロック管理を処理し、サーバと通信するクライアント状態マネージャ（ＣＳＭ）が含まれる。フラッシュ・コピーの展望から、ＩＦＳは、下記の動作についてＣＳＭに接触する。
・読取　ＣＳＭ　ＡＰＩ、csmTranslateBlocks()を使用して、仮想－物理変換を得る。
・書込　下記の２ステップで行われる
１．　ＩＦＳは、ページ・キャッシュへの書込を受け入れる前に、バッキング・ブロックを有することを確かめる必要がある。したがって、ＩＦＳによって、ＡＰＩ、csmAttachBlocks()を介してＣＳＭが呼び出される。
２．　csmAttachBlocksの成功の際に、ＩＦＳによって書込を完了できるようになる。
・DirectIO：この場合に、書込は直接にディスクに対して行われる。
・BufferedIO：この場合に、ＩＦＳは、ページをキャッシュに取り込み、それを修正し、その後、ディスクに書き込む。
・切捨　ＣＳＭ　ＡＰＩ、csmDetachBlocks()を使用してファイルを縮小する。
【００４６】
ＩＦＳによって、ファイル・ブロック関連の操作を行うために、下記の３つのインターフェースが使用される。
１．　csmAttachBlocks()：ＩＦＳの意図が、write()である時に、このインターフェースが使用される。ＣＳＭは、そのキャッシュを介してこの要求を満足できない場合に、型stpMsgType_BlkDiskAllocateのトランザクションをサーバに送る。
２．　csmTranslateBlocks()：このインターフェースは、read()またはwrite()のいずれかに使用される。ＩＦＳは、読取中およびキャッシュのハードニング中（書込動作の一部）に、このインターフェースを使用することができる。ＣＳＭは、そのキャッシュを介してこの要求を満足できない場合に、型stpMsgType_BlkDiskGetSegmentのトランザクションをサーバに送る。
３．　csmDetachBlocks()：これは、truncate()に使用される。
【００４７】
本発明の態様によれば、２タイプの仮想－物理マッピングが、ＣＳＭで維持される。
１．読取変換：これによって、読取用の仮想－物理マッピングが指定される。
２．書込変換：これによって、書込用の仮想－物理マッピングが指定される。
【００４８】
一実施形態で、セグメントが、下記の３つの状態の読取および書込の変換リストを有する場合がある。
・有効な読取変換と無効な書込変換。
・有効な書込変換と無効な読取変換。
・読取変換と書込変換の両方が有効である。

【００４９】
読取
・読取システム呼出しについて、ＩＦＳは、読取フラグをセットしてcsmTranslateBlocks()を呼び出して、読取変換が必要であることを示す。
・「読取」変換について、ＣＳＭは、書込変換を先に調べる。書込変換が存在し、使用中である場合には、書込ブロック変換を返す。
・そうでない場合には、ＣＳＭは、読取変換を調べる。読取変換が使用可能であり、使用中である場合には、その読取変換が返される。
・そうでない場合には、ＣＳＭは、これらのブロックを０で充てんする必要があることを示す、０を返す。
【００５０】
書込
・書込システム呼出しについて、ＩＦＳは、csmAttachBlocks()を呼び出す。成功の場合に、バッキング・ブロックが割り振られていることが保証される。
・更新の場合には、ＩＦＳは、そのブロックをキャッシュに戻す必要がある。したがって、PageInスレッドによって、読取フラグ付きでcsmTranslateBlocks()を呼び出す。上の「読取」の論理から、ＩＦＳはブロック変換を得る（新規ブロックへの書込について、このステップをスキップすることができる）。
・ここで、ＩＦＳは、キャッシュ内ページを更新し、そのページをディスクにフラッシュする準備ができたならば、書込フラグ付きでcsmTranslateBlocks()をもう一度呼び出す。ここで、ＣＳＭは、書込変換だけを与える必要がある。
・ＩＦＳは、前のステップで与えられた変換を使用し、キャッシュをフラッシュする（ディスクに書き込む）。
【００５１】
切捨
・ＩＦＳが、ファイル縮小のためにcsmDetachBlocks()を呼び出す。
・ＣＳＭは、対応する仮想ブロックの読取変換および書込変換の両方を、無効状態にマークする必要がある。
【００５２】
上の動作は、ＣＳＭのキャッシュで行われる。定期的な間隔で、または指定された状態で、ＣＳＭは、blkdisk更新を介する修正によってサーバを更新することができる。また、上の説明では、主として、書込が「キャッシング付き／バッファ付き入出力」であると仮定される。「直接入出力」書込について、多少の差違がある。
【００５３】
直接入出力
直接入出力は、必ずしもブロック境界で行われないので、write()の位置合せされない部分についてキャッシュ入出力を模倣する必要が生じる場合がある。
【００５４】
stpMsgType_BlkDiskAllocateまたはstpMsgType_BlkDiskGetSegmentのいずれかに応答して、ＣＳＭが、要求される各セグメントのエクステントのリストを得る。これらのセグメント変換は、アンマーシャルされ、ＣＳＭのキャッシュに保管される。
【００５５】
サーバは、書込変換と正確に同一である場合に読取変換を送らないようにすることができる。これが最適化になる可能性がある。
【００５６】
各セグメントは、mcBlkDiskSegmentという名前のデータ構造体によって表される。
【００５７】
このキャッシュ内セグメント構造体の要素に、下記を含めることができる。
【００５８】

【表１】

s_objP	このセグメントが属するファイル･オブジェクトへのポインタ。
s_segNo	ファイル内のセグメント番号。
s_readExtentCount	このセグメント内の読取エクステントの現在の個数。
s_readExtentList	このセグメントの「読取」変換を表す連続するブロック・セグメントのリスト。
s_inlineReadExtent	上のリストのインライン表現。
s_readBlockUsedState	ビット・マップである。セグメント内のブロックごとに１ビット。ブロックが使用中／未使用状態のどちらであるかを示す。１－使用中、０－未使用。
s_writeExtentCount	このセグメントの書込エクステントの現在の個数。
s_writeExtentList	このセグメントの「書込」変換を表す連続するブロック・セグメントのリスト。
s_writeBlockUsedStat	ビット・マップである。セグメント内のブロックごとに１ビット。ブロックが使用中／未使用状態のどちらであるかを示す。１－使用中、０－未使用。
s_isDirty	ライブ・ブロック状態ビット・ベクトルに、サーバと同期化されなければならない更新が含まれる場合に真。
s_extentListValid	エクステント・リストが有効であり、キャッシュ内にある場合に真。

【００５９】
クライアントは、排他データ・ロックの下でs_readBlockUsedStateおよびs_writeBlockUsedStateだけを変更する。セグメント変換の残りの部分は、クライアント側では変更されない。したがって、更新を送っている間に、クライアントは、この２つのビット・マップだけをサーバに送信する。
【００６０】
読取動作では、ビット・マップが全く変更されない、すなわち、s_readBlockUsedStateまたはs_writeBlockUsedStateのいずれかが、読取変換をもたらすのに使用されるが、変更されない。
【００６１】
書込動作では、s_writeBlockUsedStateだけが操作され、s_readBlockUsedStateは使用されない。読取動作とは異なって、s_writeBlockUsedStateのビットマップを変更（セットのみ）して、成功裡の書込動作を示す場合がある。
【００６２】
切捨動作では、両方のビット・マップが変更される可能性がある。
【００６３】
したがって、短く言えば、読取動作では何も変更されず、書込動作ではs_writeBlockUsedStateビットマップ・ベクトルの少数のビットがセットされる場合があり、切捨動作では両方のビットマップ・ベクトルのビットがアンセットされる場合がある。
【００６４】
要約すると、キャッシュ付き入出力の場合に、コピー・オン・ライト（ＣＯＷ）を、ほとんどコストなしで達成することができる。これは、通常、書込動作の完了にかかわる下記の２つの異なるスレッド／動作があるからである。

【００６５】
１．ターゲット・データをキャッシュ内で更新／変更させるPageInスレッド／動作と、
【００６６】
２．更新されたページをディスクにフラッシュ・バックするPageOutスレッド／動作。
【００６７】
これを与えられて、ＣＯＷを、PageInに読取変換、PageOutに書込変換を使用することによって実行することができる。
【００６８】
例として下記を検討されたい。
【００６９】
新しいファイルの作成の際に、ＣＳＭは、
－　書込ブロック・エクステント
－　ＮＵＬＬ読取ブロック・エクステント
を得る。
【００７０】
新しいファイルなので、ページ・インすべきものはなく、したがって、書込データは、キャッシュの空白ページに送られる。
【００７１】
書込が進行する際に、ＩＦＳは、（ＣＳＭインターフェースを介して）s_writeBlockUsedStateビット・ベクトルの対応するビットをセットし、それらが使用中であることを示す。
【００７２】
上で説明したように、このブロック範囲内の将来の読取および書込では、書込エクステントからの変換を得る。
【００７３】
管理者が、フラッシュ・コピーをとると仮定する。
【００７４】
フラッシュ・コピー動作の一部として、サーバによって、クライアントからのすべてのデータ・ロックが取り消される。したがって、修正されたデータが、ディスクに同期化され、メタデータが、更新トランザクションを介してサーバに送られる。クライアントは、データ・ロックを有しないので、その変換のすべてが無効になる。
【００７５】
フラッシュ・コピーの後
読取について、クライアントは、csmTranslateBlocks()を呼び出し、サーバは、「読取」変換を呼び出すことができるが、「書込」変換はＮＵＬＬになる。
【００７６】
書込について、クライアントは、csmAttachBlocks()を呼び出す。サーバは、読取変換を返し（上と同一）、書込エクステント・リストについて、サーバは、未使用ブロックの新しい組を割り振り、返さなければならない。したがって、クライアントは、２つの変換を有する。クライアントは、「ページイン」の一部として読取変換を使用し、「ページアウト」の一部として書込変換を使用する。
【００７７】
したがって、フラッシュ・コピーの一部であったデータ・ブロックを、バッファ・キャッシュにPageInし、更新を適用する。ページ・アウト中に、書込変換を使用するが、この書込は、新しい物理ブロックへのPageOutスレッドをポイントする。
【００７８】
directI0のＣＯＷは、多少異なる。直接入出力について：
ｏ　ＩＦＳは、書込フラグをセットしてcsmTranslateBlocks()を呼び出して、書込を進めるためのバッキング・ブロックを有するかどうかを調べる。ＣＳＭからの変換を分析した後に、バッキング・ブロックが割り振られていない場合に、ＩＦＳは、csmAttachBlock

s()を呼び出す。
　o　csmAttachBlocks()について、ＩＦＳは、そのデータロックをＳＨＡＲＥＤ＿ＷＲＩＴＥモードからＥＸＣＬＵＳＩＶＥに切り替える必要がある。
　o　書込が終了した後に、クライアントは、これらのブロックをＵＳＥＤ状態にマークする。この時に、クライアントは、排他モードのデータ・ロックを保持する必要がある。
　o　直接入出力境界が、ブロック・サイズに位置合わせされていない場合に、最初のブロックおよび最後のブロックへの入出力が、キャッシング付きの形で行われる。これを行うステップに、下記が含まれる。
　　・「読取変換」についてＣＳＭを呼び出す。
　　・ローカル・カーネル・バッファを割り振る。
　　・新たに割り振られたカーネル・バッファにディスク・ブロックを読み込む。
　　・カーネル・バッファを更新する。
　　・書込変換についてＣＳＭを呼び出す。
　　・新しいブロック位置に書き込む（書込変換）
　　・ＣＳＭを呼び出して新しい書込ブロックの「ＵＳＥＤ」ビットをマークする。
【００７９】
　中間ブロックについて、
　－　境界をブロック・サイズに位置合せする。
　－　ＣＳＭを呼び出して、書込変換を得る。
　－　ディスクに書き込む。
　－　ＣＳＭを呼び出して、ブロックの「ＵＳＥＤ」ビットをマークする。
【００８０】
　下記は、クライアントでのビット・メトリックスの例である。
【００８１】
　WriteBlockUsedStateビット配列：Wbit
【００８２】
　ReadBlockUsedStateビット配列：Rbit
【００８３】
　かぎ括弧内で、（@＜＞）は、物理ブロック・アドレスを示す。すなわち（@Ａ）は、物理ブロック・アドレスがＡであることを意味する。
【００８４】
　ＵＤ　－　未定義
【００８５】

【表２】

	仮想ブロック#1	仮想ブロック#2	仮想ブロック#3	仮想ブロック#4

オペランド	(W Phy Blk @) Wbit	(R Phy Blk @) Rbit	(W Phy Blk @) Wbit	(R Phy Blk @) Rbit	(W Phy Blk @) Wbit	(R Phy Blk @) Rbit	(W Phy Blk @) Wbit	(R Phy Blk @) Rbit
ファイル作成および割振	(@A) 0	UD UD	(@B) 0	UD UD	(@C) 0	UD UD	(@D) 0	UD UD
３ブロック書込	(@A) 1	UD UD	(@B) 1	UD UD	(@C) 1	UD UD	(@D) 0	UD UD
ファイル読取(上から変更なし)	(@A) 1	UD UD	(@B) 1	UD UD	(@C) 1	UD UD	(@D) 0	UD UD
２ブロックに切捨	(@A) 1	UD UD	(@B) 1	UD UD	(@C) 0	UD UD	(@D) 0	UD UD
読取	NULL UD	(@ A) 1	NULL UD	(@B) 1	UD UD	NULL UD	UD UD	NULL UD
最初の２ブロックの更新を希望	(@X) 0	(@A) 1	(@Y) 0	(@B) 1	UD UD	NULL UD	UD UD	NULL UD
最初の２ブロックを書込（更新）	(@X) 1	(@A) 1	(@Y) 1	(@B) 1	UD UD	NULL UD	UD UD	NULL UD
１ブロックに切捨	(@X) 1	(@A) 1	(@Y) 0	(@B) 0	UD UD	NULL UD	UD YD	NULL UD

【００８６】
クライアントは、クライアント・サーバ環境で、WriteBlockUsedStateビット配列およびReadBlockUsedStateビット配列だけをサーバに送信する。
【００８７】
ＣＳＭは、これらのビットを解釈し、下記のようにマッピングする
【００８８】

【表３】

X = このブロックに関するマッピングなし
M = このブロックに関するマッピングが存在する

状態	書込マッピング	書込ビット	読取マッピング	読取ビット
未割振り	X	0	X	0
未定義	X	0	X	1
未定義	X	0	M	0
読取可能、共用（COW ブロック割振りが必要）	X	0	M	1
未定義	X	1	X	0
未定義	X	1	X	1
未定義	X	1	M	0
未定義	X	1	M	1
書込可能だが読取不能	M	0	X	0
未定義	M	0	X	1
未定義	M	0	M	0
COW 保留中、読取で RM、書込で WM を使用	M	0	M	1
書込マッピングを介して書込可能かつ読取可能	M	1	X	0
未定義	M	1	X	1
未定義	M	1	M	0
未定義	M	1	M	1

【００８９】
ブロックのマッピングを判定するために、ＣＳＭは、まず、書込マッピングを調べ、書込マッピングが存在し、対応するＷビットがセットされている場合に、そのマッピングを、読取動作および書込動作の両方に使用する。
【００９０】
書込マッピングが存在するが、Ｗビットが０である場合には、ＣＳＭは、読取マッピングを探す。
【００９１】
読取マッピングが存在しない場合には、そのブロックは未使用（未初期化）とみなされ、そのブロックは、書込動作だけに使用することができる。すべての読取動作によって、バッファに０が充てんされる。
【００９２】
読取マッピングが存在する場合には、そのブロックは、使用中（初期化済み）とみなされ、ＣＳＭは、Ｒビットが１であると仮定する（そうでなければならない）。この状態のブロックは、ＣＯＷ保留中とのみみなされ、その内容を修正するにはＣＯＷが必要である。
【００９３】
書込マッピングが存在しない場合には、ＣＳＭは、読取マッピングを探す。読取マッピングが存在する場合に、ＣＳＭは、Ｒビットが１であると仮定し（そうでなければならない）、そのブロックは、読取動作に使用中（初期化済み）だが書込動作には使用されていないとみなされる。書込動作を実行する前に、ＣＭＳは、サーバによって新しいバッキン

グ・ブロックが割り振られることを要求しなければならない。

【００９４】

ＣＳＭは、サーバにビット・ベクトルを返して、使用中、切捨済み、またはコピー・オン・ライト済みあるいはこれらの組合せとしてブロックの状況の変化を示す。リターンの際に、サーバは、下記のようにビットを解釈する。

【００９５】

【表４】

Ｘ＝ドントケア

状態	W	R
ブロックはライブである（ＣＯＷまたは使用中の割り振られたブロックをハードニング）	1	X
このブロックを切り捨てる（使用中の場合に未使用としてマークされる－解放可能）	0	0
このブロックに対する変更なし（ＣＯＷ保留を取り消すことができる）	0	1

【００９６】

書込ビットがセットされている場合には：

サーバは、読取ビットを無視する。

ブロックがAllocated状態であった（すなわち、従来の形で割り振られたがまだライブでない）場合には、Live状態に変更される（すなわち、ブロックが書込可能かつ読取可能）。

ブロックがCOW_Pending状態であった（すなわち、ブロックが、読取マッピングおよび書込マッピングの両方を有し、これらがＣＯＷにおいて異なる）場合には、Live状態になる（すなわち、ブロックが、書込マッピングであったものを介して書込可能かつ読取可能）。

【００９７】

書込ビットがセットされず、読取ビットがセットされる場合には：

これによって、問題のブロックに関する変更がないことがサーバに示される。

ブロックが、Allocated、COW_Pending、またはPIT_COW_Pending状態であった場合には、そのブロックは、この状態に留まることができ、あるいは、そのブロックを非同期に解放することができる。クライアントは、そのブロックが解放されることまたはCOW_Pending状態の１つに留まるかどうかを仮定することができない。

【００９８】

書込ビットがセットされず、読取ビットがセットされない場合には：

これによって、ブロックが切り捨てられた（前に割り振られたと仮定して）ことがサーバに示される。

ブロックがUnallocated状態であった婆には、そのブロックはUnallocated状態のままになる。

ブロックがAllocated状態であった場合には、そのブロックはAllocated状態のままになる。

ブロックがLive状態であった場合には、そのブロックはAllocated状態になる。

ブロックがShared状態であった場合には、そのブロックはUnallocated状態になり、読取（専用）マッピングが破棄される。

ブロックがCOW_Pending状態であった場合には、そのブロックはAllocated状態になる。

ブロックがPIT_COW_Pending状態であった場合には、そのブロックはAllocated状態になる。

【００９９】

長所

有利なことに、本明細書で、冗長な入出力を最小限にすることによって、従来のデータ・ファイル書込処理に対する最小限の追加コストで、コンピューティング環境内でコピー・オン・ライトを実施する技法が提示される。提示されたコピー・オン・ライト技法は、標準ファイルシステム・ドライバなどの上位層に透過的である。この技法には、２つの異なる同時変換すなわち、読取マッピング・テーブルおよび書込マッピング・テーブルを使用して、単一の書込動作を使用してファイル内のデータの単位のコピー・オン・ライトを達成することが含まれる。もう１つの態様では、クライアント・サーバ環境の複数のクライアントにまたがるファイルの分散コピー・オン・ライトを実施する技法が提示される。有利なことに、この分散コピー・オン・ライト実施形態では、中央サーバへの負荷が減り、クライアントの追加に伴ってスケーリングされ、１つまたは複数のクライアントが動作不能になる場合があるにもかかわらず、ファイルのコピー・オン・ライトを進行できるようになる。さらに、提示された分散コピー・オン・ライトでは、並列コピー・オン・ライトが可能であり、作業負荷を複数のクライアントの間で分散でき、これによって、リソースのより効率的な利用がもたらされる。

【０１００】

代替実施形態

コンピューティング環境の例を提供したが、これらは例にすぎない。他の実施形態を使用することができる。たとえば、本明細書で、ファイルシステムに関する例を説明したが、これは単なる一例にすぎない。本発明の１つまたは複数の他の態様は、他の環境に適用可能である。

【０１０１】

本発明を、たとえばコンピュータ使用可能媒体を有する製造品（たとえば、１つまたは複数のコンピュータ・プログラム製品）に含めることができる。この媒体は、その中に、たとえば、本発明の機能を提供し促進するコンピュータ可読プログラム・コード手段または論理（たとえば、命令、コード、コマンドなど）を実施される。製造品を、コンピュータ・システムの一部として含めるか、別々に販売することができる。

【０１０２】

さらに、本発明の機能を実行するために機械によって実行可能な命令の少なくとも１つのプログラムを実施する、機械によって可読の少なくとも１つのプログラム記憶装置を提供することができる。

【０１０３】

本明細書で示された流れ図は、例にすぎない。本発明の趣旨から逸脱しない、これらの図または図に示されたステップ（または動作）に対する多数の変形形態がありえる。たとえば、ステップを、異なる順序で実行することができ、ステップを追加、削除、または修正することができる。これらの変形形態のすべてが、請求される発明の一部とみなされる。

【０１０４】

好ましい実施形態を図示し、本明細書で詳細に説明したが、本発明の趣旨から逸脱せずに、さまざまな修正、追加、置換、および類似物を行うことができ、したがって、これらが、請求項で定義される発明の趣旨に含まれるとみなされることを、当業者は諒解するであろう。

【０１０５】

まとめとして、本発明の構成に関して以下の事項を開示する。

【０１０６】

（１）コンピューティング環境でコピー・オン・ライトを実施する方法であって、

（ｉ）ファイルのデータのブロックを修正のために物理記憶装置から読み取るのに使用される第１仮想ブロック－物理ブロック・マッピングを実行するのに第１マッピング・テーブルを使用することと、

（ｉｉ）前記ファイルの前記データの修正されたブロックを物理記憶装置に書き込むの

に使用される第２仮想ブロック－物理ブロック・マッピングを実行するのに第２マッピング・テーブルを使用することであって、データの前記ブロックのコピー・オン・ライトが、単一の書込動作を使用して達成される、使用することと
を含む方法。
（２）前記第１マッピング・テーブルが、読取マッピング・テーブルを含み、前記第２マッピング・テーブルが、書込マッピング・テーブルを含み、前記書込マッピング・テーブルが、前記読取マッピング・テーブルの対応する仮想ブロック－物理ブロック・マッピングと異なる、前記物理記憶装置の物理ブロックにマッピングされる少なくとも１つの仮想ブロックを含む、上記（１）に記載の方法。
（３）前記コピー・オン・ライト実施形態が、さらに、まず修正が部分ブロック書込または全ブロック書込のどちらを含むかを判定することと、部分ブロック書込の場合に、前記使用すること（ｉ）および前記使用すること（ｉｉ）を実行し、そうでない場合に前記使用すること（ｉ）を実行せずに前記実行すること（ｉｉ）を実行することとを含む、上記（１）に記載の方法。
（４）前記実行すること（ｉ）が、データの前記ブロックを物理記憶装置からバッファに読み取ることを含み、前記方法が、さらに、前記使用すること（ｉｉ）を実行する前に前記バッファ内のデータの前記ブロックを修正することを含む、上記（１）に記載の方法。
（５）前記コンピューティング環境が、ファイルシステム・サーバおよび少なくとも１つのクライアントを含むクライアント・サーバ環境を含み、前記使用すること（ｉ）および前記使用すること（ｉｉ）が、前記クライアント・サーバ環境の前記少なくとも１つのクライアントによって実行される、上記（１）に記載の方法。
（６）前記少なくとも１つのクライアントによって実行される前記使用すること（ｉ）および前記使用すること（ｉｉ）が、前記ファイルのデータの前記ブロックのコピー・オン・ライトを実行する時に、前記第１マッピング・テーブルおよび前記第２マッピング・テーブルの少なくとも１つを入手するために前記ファイルシステム・サーバへの少なくとも１つの呼出しを行うことをさらに含む、上記（５）に記載の方法。
（７）データの前記修正されたブロックを物理記憶装置に書き込んだ後に、前記第１マッピング・テーブルを更新することをさらに含み、前記更新することが、前記第１マッピング・テーブルの少なくとも１つの仮想ブロック－物理ブロック変換を、前記第２マッピング・テーブルの対応する仮想ブロック－物理ブロック変換と一致するように修正することを含む、上記（５）に記載の方法。
（８）前記コンピューティング環境が、コンピューティング・ユニットおよび外部記憶ユニットを含み、前記外部記憶ユニットが、前記物理記憶装置を含み、前記使用すること（ｉ）および前記使用すること（ｉｉ）が、前記コンピューティング・ユニットによって実行される、上記（１）に記載の方法。
（９）クライアント・サーバ・コンピューティング環境でコピー・オン・ライトを容易にする方法であって、
前記クライアント・サーバ・コンピューティング環境のファイルシステム・サーバで、ファイルに関する読取マッピング・テーブルおよび書込マッピング・テーブルを維持すること
を含み、前記読取マッピング・テーブルが、修正のために物理記憶装置から前記ファイルのデータのブロックを読み取るのに使用される第１仮想ブロック－物理ブロック・マッピングを実行するのに使用可能であり、前記書込マッピング・テーブルが、前記ファイルの前記データの修正されたブロックを物理記憶装置に書き込むのに使用される第２仮想ブロック－物理ブロック・マッピングを実行するのに使用可能であり、前記読取マッピング・テーブルおよび前記書込マッピング・テーブルを使用して、データのブロックのコピー・オン・ライトが単一の書込動作を使用して達成可能である
方法。
（１０）前記ファイルに関する前記書込マッピング・テーブルが、前記読取マッピング・テーブルの対応する仮想ブロック－物理ブロック変換と異なる、前記物理記憶装置の物理

ブロックにマッピングされる少なくとも１つの仮想ブロックを含む、上記（９）に記載の方法。
（１１）前記ファイルのデータのブロックのコピー・オン・ライトが実行された後に前記読取マッピング・テーブルを更新することをさらに含み、前記更新することが、前記読取マッピング・テーブルの少なくとも１つの仮想ブロック－物理ブロック変換を、前記書込マッピング・テーブルの対応する仮想ブロック－物理ブロック変換と一致するように修正することを含む、上記（９）に記載の方法。
（１２）複数のクライアントを有するクライアント・サーバ環境内でファイルのコピー・オン・ライトを実施する方法であって、
前記クライアント・サーバ環境の複数のクライアントを使用してファイルのコピー・オン・ライトを実行することであって、
（ｉ）前記複数のクライアントの第１クライアントによって、前記ファイルのデータの少なくとも１つのブロックのコピー・オン・ライトを実行することと、
（ｉｉ）前記複数のクライアントの第２クライアントによって、前記ファイルのデータの少なくとも１つの他のブロックのコピー・オン・ライトを実行することと
を含む、実行すること
を含む方法。
（１３）前記実行すること（ｉ）が、前記第１クライアントによって、単一の書込動作を使用して前記ファイルのデータの前記少なくとも１つのブロックのコピー・オン・ライトを実行することを含み、前記実行すること（ｉｉ）が、前記第２クライアントによって、単一の書込動作を使用してデータの前記少なくとも１つの他のブロックのコピー・オン・ライトを実行することを含む、上記（１２）に記載の方法。
（１４）前記クライアント・サーバ環境のファイルシステム・サーバが、前記ファイルを含む少なくとも１つの共用記憶装置に関連し、前記ファイルシステム・サーバが、前記少なくとも１つの共用記憶装置に保管された前記ファイルに関するマッピング・テーブルを維持し、前記実行すること（ｉ）が、前記第１クライアントによって、前記ファイルシステム・サーバから前記ファイルに関する読取マッピング・テーブルおよび書込マッピング・テーブルを入手することと、前記ファイルのデータの前記少なくとも１つのブロックの前記コピー・アンド・ライトを実行する際に前記読取マッピング・テーブルおよび前記書込マッピング・テーブルを使用することとを含み、前記実行すること（ｉｉ）が、前記第２クライアントによって、前記ファイルシステム・サーバから前記ファイルに関する前記読取マッピング・テーブルおよび前記書込マッピング・テーブルを入手することと、前記ファイルのデータの前記少なくとも１つの他のブロックの前記コピー・アンド・ライトを実行する際に前記読取マッピング・テーブルおよび前記書込マッピング・テーブルを使用することとを含む、上記（１３）に記載の方法。
（１５）前記第１クライアントによって、前記第１クライアントが前記ファイルのデータの前記少なくとも１つのブロックのコピー・オン・ライトを実行したことを前記ファイルシステム・サーバに知らせることと、それに応答して、前記ファイルシステム・サーバによって維持される前記ファイルに関する前記読取マッピング・テーブルおよび前記書込マッピング・テーブルの少なくとも１つを更新することとをさらに含む、上記（１４）に記載の方法。
（１６）前記ファイルシステム・サーバが、前記第１クライアントによってコピー・オン・ライト更新された前記ファイルのデータの前記少なくとも１つのブロックおよび前記第２クライアントによってコピー・オン・ライト更新された前記ファイルのデータの前記少なくとも１つの他のブロックに対するコピー・オン・ライトを前記複数のクライアントのすべてのクライアントが行えなくする、上記（１４）に記載の方法。
（１７）前記クライアント・サーバ環境のファイルシステム・サーバが、前記ファイルを含む少なくとも１つの共用記憶装置に関連し、前記方法が、さらに、前記ファイルの前記コピー・オン・ライトを開始するために前記ファイルシステム・サーバによって前記ファイルを制御することを含み、前記開始することが、前記実行すること（ｉ）および前記実

行すること（ｉｉ）によって使用される前記ファイルに関する書込マッピング・テーブルを更新することを含む、上記（１２）に記載の方法。
（１８）クライアント・サーバ環境内でファイルのコピー・オン・ライトを容易にする方法であって、
前記クライアント・サーバ環境の共用記憶ユニットに保管されたファイルのコピー・オン・ライトの実施をファイルシステム・サーバから制御することであって、前記制御することが、前記クライアント・サーバ環境の第１クライアントが前記ファイル内のデータの部分をコピー・オン・ライトできるようにすることと、前記クライアント・サーバ環境の第２クライアントが前記ファイル内の前記データの異なる部分をコピー・オン・ライトできるようにすることとを含み、前記ファイルシステム・サーバが、前記ファイルの分散コピー・オン・ライトの実行を制御し、容易にする、制御すること
を含む方法。
（１９）前記制御することが、前記ファイルのコピー・オン・ライトを開始するために前記ファイルシステム・サーバによって前記ファイルを制御することを含み、前記開始することが、前記コピー・オン・ライトを実行するのに使用される前記ファイルに関する少なくとも１つのマッピング・テーブルを更新することを含む、上記（１８）に記載の方法。
（２０）前記ファイルシステム・サーバで、前記ファイルに関する読取マッピング・テーブルおよび書込マッピング・テーブルを維持することをさらに含み、前記読取マッピング・テーブルおよび前記書込マッピング・テーブルが、前記コピー・オン・ライトの実行に使用される、上記（１８）に記載の方法。
（２１）前記コピー・オン・ライトの実行の後に、前記ファイルに関する前記読取マッピング・テーブルおよび前記書込マッピング・テーブルの少なくとも１つを更新することをさらに含む、上記（２０）に記載の方法。
（２２）前記制御することが、前記ファイルシステム・サーバによって、前記ファイルの前記コピー・オン・ライトの一部としての、前記第１クライアントによってコピー・オン・ライト更新された前記ファイル内の前記データの前記部分に対する追加更新および前記第２クライアントによってコピー・オン・ライト更新された前記ファイル内の前記データの前記異なる部分に対する追加更新を防ぐことをさらに含む、上記（１８）に記載の方法。
（２３）コンピューティング環境でコピー・オン・ライトを実施するシステムであって、
（ｉ）ファイルのデータのブロックを修正のために物理記憶装置から読み取るのに使用される第１仮想ブロック－物理ブロック・マッピングを実行するのに第１マッピング・テーブルを使用する手段と、
（ｉｉ）前記ファイルの前記データの修正されたブロックを物理記憶装置に書き込むのに使用される第２仮想ブロック－物理ブロック・マッピングを実行するのに第２マッピング・テーブルを使用する手段であって、データの前記ブロックのコピー・オン・ライトが、単一の書込動作を使用して達成される、使用する手段と
を含むシステム。
（２４）前記第１マッピング・テーブルが、読取マッピング・テーブルを含み、前記第２マッピング・テーブルが、書込マッピング・テーブルを含み、前記書込マッピング・テーブルが、前記読取マッピング・テーブルの対応する仮想ブロック－物理ブロック・マッピングと異なる、前記物理記憶装置の物理ブロックにマッピングされる少なくとも１つの仮想ブロックを含む、上記（２３）に記載のシステム。
（２５）前記コピー・オン・ライト実施形態が、さらに、まず修正が部分ブロック書込または全ブロック書込のどちらを含むかを判定し、部分ブロック書込の場合に、前記使用すること（ｉ）および前記使用すること（ｉｉ）を実行し、そうでない場合に前記使用すること（ｉ）を実行せずに前記実行すること（ｉｉ）を実行する手段を含む、上記（２３）に記載のシステム。
（２６）前記実行する手段（ｉ）が、データの前記ブロックを物理記憶装置からバッファに読み取る手段を含み、前記システムが、さらに、前記使用すること（ｉｉ）を実行する

前に前記バッファ内の前記ブロックを修正する手段を含む、上記（２３）に記載のシステム。
（２７）前記コンピューティング環境が、ファイルシステム・サーバおよび少なくとも１つのクライアントを含むクライアント・サーバ環境を含み、前記使用する手段（ｉ）および前記使用する手段（ｉｉ）が、前記クライアント・サーバ環境の前記少なくとも１つのクライアントによって実行される、上記（２３）に記載のシステム。
（２８）前記少なくとも１つのクライアントによって実行される前記使用する手段（ｉ）および前記使用する手段（ｉｉ）が、前記ファイルのデータの前記ブロックのコピー・オン・ライトを実行する時に、前記第１マッピング・テーブルおよび前記第２マッピング・テーブルの少なくとも１つを入手するために前記ファイルシステム・サーバへの少なくとも１つの呼出しを行う手段をさらに含む、上記（２７）に記載のシステム。
（２９）データの前記修正されたブロックを物理記憶装置に書き込んだ後に、前記第１マッピング・テーブルを更新する手段をさらに含み、前記更新する手段が、前記第１マッピング・テーブルの少なくとも１つの仮想ブロック－物理ブロック変換を、前記第２マッピング・テーブルの対応する仮想ブロック－物理ブロック変換と一致するように修正する手段を含む、上記（２７）に記載のシステム。
（３０）前記コンピューティング環境が、コンピューティング・ユニットおよび外部記憶ユニットを含み、前記外部記憶ユニットが、前記物理記憶装置を含み、前記使用する手段（ｉ）および前記使用する手段（ｉｉ）が、前記コンピューティング・ユニットによって実行される、上記（２３）に記載のシステム。
（３１）クライアント・サーバ・コンピュータ環境でコピー・オン・ライトを容易にするシステムであって、
前記クライアント・サーバ・コンピューティング環境のファイルシステム・サーバで、ファイルに関する読取マッピング・テーブルおよび書込マッピング・テーブルを維持する手段
を含み、前記読取マッピング・テーブルが、修正のために物理記憶装置から前記ファイルのデータのブロックを読み取るのに使用される第１仮想ブロック－物理ブロック・マッピングを実行するのに使用可能であり、前記書込マッピング・テーブルが、前記ファイルの前記データの修正されたブロックを物理記憶装置に書き込むのに使用される第２仮想ブロック－物理ブロック・マッピングを実行するのに使用可能であり、前記読取マッピング・テーブルおよび前記書込マッピング・テーブルを使用して、データのブロックのコピー・オン・ライトが単一の書込動作を使用して達成可能である
システム。
（３２）前記ファイルに関する前記書込マッピング・テーブルが、前記読取マッピング・テーブルの対応する仮想ブロック－物理ブロック変換と異なる、前記物理記憶装置の物理ブロックにマッピングされる少なくとも１つの仮想ブロックを含む、上記（３１）に記載のシステム。
（３３）前記ファイルのデータのブロックのコピー・オン・ライトが実行された後に前記読取マッピング・テーブルを更新する手段をさらに含み、前記更新する手段が、前記読取マッピング・テーブルの少なくとも１つの仮想ブロック－物理ブロック変換を、前記書込マッピング・テーブルの対応する仮想ブロック－物理ブロック変換と一致するように修正する手段を含む、上記（３１）に記載のシステム。
（３４）複数のクライアントを有するクライアント・サーバ環境内でファイルのコピー・オン・ライトを実施するシステムであって、
（ｉ）前記クライアント・サーバ環境の第１クライアントで、コピー・オン・ライトされる前記ファイルのデータの少なくとも１つのブロックのコピー・オン・ライトを実行する手段と、
（ｉｉ）前記クライアント・サーバ環境の第２クライアントで、コピー・オン・ライトされる前記ファイルのデータの少なくとも１つの他のブロックのコピー・オン・ライトを実行する手段であって、前記ファイルの前記コピー・オン・ライトの異なる部分が、前記

クライアント・サーバ環境内の前記複数のクライアントの異なるクライアントによって実行される、手段と
を含むシステム。
（３５）前記実行する手段（ｉ）が、前記第１クライアントによって、単一の書込動作を使用して前記ファイルのデータの前記少なくとも１つのブロックのコピー・オン・ライトを実行する手段を含み、前記実行する手段（ｉｉ）が、前記第２クライアントによって、単一の書込動作を使用してデータの前記少なくとも１つの他のブロックのコピー・オン・ライトを実行する手段を含む、上記（３４）に記載のシステム。
（３６）前記クライアント・サーバ環境のファイルシステム・サーバが、前記ファイルを含む少なくとも１つの共用記憶装置に関連し、前記ファイルシステム・サーバが、前記少なくとも１つの共用記憶装置に保管された前記ファイルに関するマッピング・テーブルを維持し、前記実行する手段（ｉ）が、前記第１クライアントによって、前記ファイルシステム・サーバから前記ファイルに関する読取マッピング・テーブルおよび書込マッピング・テーブルを入手し、前記ファイルのデータの前記少なくとも１つのブロックの前記コピー・アンド・ライトを実行する際に前記読取マッピング・テーブルおよび前記書込マッピング・テーブルを使用する手段を含み、前記実行する手段（ｉｉ）が、前記第２クライアントによって、前記ファイルシステム・サーバから前記ファイルに関する前記読取マッピング・テーブルおよび前記書込マッピング・テーブルを入手し、前記ファイルのデータの前記少なくとも１つの他のブロックの前記コピー・アンド・ライトを実行する際に前記読取マッピング・テーブルおよび前記書込マッピング・テーブルを使用する手段を含む、上記（３５）に記載のシステム。
（３７）前記第１クライアントによって、前記第１クライアントが前記ファイルのデータの前記少なくとも１つのブロックのコピー・オン・ライトを実行したことを前記ファイルシステム・サーバに知らせ、それに応答して、前記ファイルシステム・サーバによって維持される前記ファイルに関する前記読取マッピング・テーブルおよび前記書込マッピング・テーブルの少なくとも１つを更新する手段をさらに含む、上記（３６）に記載のシステム。
（３８）前記ファイルシステム・サーバが、前記第１クライアントによってコピー・オン・ライト更新された前記ファイルのデータの前記少なくとも１つのブロックおよび前記第２クライアントによってコピー・オン・ライト更新された前記ファイルのデータの前記少なくとも１つの他のブロックに対するコピー・オン・ライトを前記複数のクライアントのいずれもが行えなくする、上記（３６）に記載のシステム。
（３９）前記クライアント・サーバ環境のファイルシステム・サーバが、前記ファイルを含む少なくとも１つの共用記憶装置に関連し、前記システムが、さらに、前記ファイルの前記コピー・オン・ライトを開始するために前記ファイルシステム・サーバによって前記ファイルを制御する手段を含み、前記開始する手段が、前記実行する手段（ｉ）および前記実行する手段（ｉｉ）によって使用される前記ファイルに関する書込マッピング・テーブルを更新する手段を含む、上記（３４）に記載のシステム。
（４０）クライアント・サーバ環境内でファイルのコピー・オン・ライトを容易にするシステムであって、
前記クライアント・サーバ環境の共用記憶ユニットに保管されたファイルのコピー・オン・ライトの実施をファイルシステム・サーバから制御する手段であって、前記制御する手段が、前記クライアント・サーバ環境の第１クライアントが前記ファイル内のデータの部分をコピー・オン・ライトできるようにし、前記クライアント・サーバ環境の第２クライアントが前記ファイル内の前記データの異なる部分をコピー・オン・ライトできるようにする手段を含み、前記ファイルシステム・サーバが、前記ファイルの分散コピー・オン・ライトの実行を制御し、容易にする、制御する手段
を含むシステム。
（４１）前記制御する手段が、前記ファイルのコピー・オン・ライトを開始するために前記ファイルシステム・サーバによって前記ファイルを制御する手段を含み、前記開始する

ことが、前記コピー・オン・ライトを実行するのに使用される前記ファイルに関する少なくとも１つのマッピング・テーブルを更新することを含む、上記（４０）に記載のシステム。
（４２）前記ファイルシステム・サーバで、前記ファイルに関する読取マッピング・テーブルおよび書込マッピング・テーブルを維持する手段をさらに含み、前記読取マッピング・テーブルおよび前記書込マッピング・テーブルが、前記コピー・オン・ライトの実行に使用される、上記（４０）に記載のシステム。
（４３）前記コピー・オン・ライトの実行の後に、前記ファイルに関する前記読取マッピング・テーブルおよび前記書込マッピング・テーブルの少なくとも１つを更新する手段をさらに含む、上記（４２）に記載のシステム。
（４４）前記制御する手段が、前記ファイルシステム・サーバによって、前記ファイルの前記コピー・オン・ライトの一部としての、前記第１クライアントによってコピー・オン・ライト更新された前記ファイル内の前記データの前記部分に対する追加更新および前記第２クライアントによってコピー・オン・ライト更新された前記ファイル内の前記データの前記異なる部分に対する追加更新を防ぐ手段をさらに含む、上記（４０）に記載のシステム。
（４５）製造品であって、
コンピューティング環境でコピー・オン・ライトを実施するコンピュータ可読プログラム・コード論理を有する少なくとも１つのコンピュータ使用可能媒体
を含み、前記コンピュータ可読プログラム・コード論理が、
（ｉ）ファイルのデータのブロックを修正のために物理記憶装置から読み取るのに使用される第１仮想ブロック－物理ブロック・マッピングを実行するのに第１マッピング・テーブルを使用する論理と、
（ｉｉ）前記ファイルの前記データの修正されたブロックを物理記憶装置に書き込むのに使用される第２仮想ブロック－物理ブロック・マッピングを実行するのに第２マッピング・テーブルを使用する論理であって、データの前記ブロックのコピー・オン・ライトが、単一の書込動作を使用して達成される、使用する論理と
を含む、製造品。
（４６）前記コンピューティング環境が、ファイルシステム・サーバおよび少なくとも１つのクライアントを含むクライアント・サーバ環境を含み、前記使用する論理（ｉ）および前記使用する論理（ｉｉ）が、前記クライアント・サーバ環境の前記少なくとも１つのクライアントによって実行される、上記（４５）に記載の製造品。
（４７）前記少なくとも１つのクライアントによって実行される前記使用する論理（ｉ）および前記使用する論理（ｉｉ）が、前記ファイルのデータの前記ブロックのコピー・オン・ライトを実行する時に、前記第１マッピング・テーブルおよび前記第２マッピング・テーブルの少なくとも１つを入手するために前記ファイルシステム・サーバへの少なくとも１つの呼出しを行うことをさらに含む、上記（４５）に記載の製造品。
（４８）前記コンピューティング環境が、コンピューティング・ユニットおよび外部記憶ユニットを含み、前記外部記憶ユニットが、前記物理記憶装置を含み、前記使用する論理（ｉ）および前記使用する論理（ｉｉ）が、前記コンピューティング・ユニットによって実行される、上記（４５）に記載の製造品。
（４９）製造品であって、
クライアント・サーバ・コンピューティング環境でコピー・オン・ライトを容易にするコンピュータ可読プログラム・コード論理を有する少なくとも１つのコンピュータ使用可能媒体
を含み、前記コンピュータ可読プログラム・コード論理が、
前記クライアント・サーバ・コンピューティング環境のファイルシステム・サーバで、ファイルに関する読取マッピング・テーブルおよび書込マッピング・テーブルを維持する論理であって、前記読取マッピング・テーブルが、修正のために物理記憶装置から前記ファイルのデータのブロックを読み取るのに使用される第１仮想ブロック－物理ブロック・

マッピングを実行するのに使用可能であり、前記書込マッピング・テーブルが、前記ファイルの前記データの修正されたブロックを物理記憶装置に書き込むのに使用される第２仮想ブロック－物理ブロック・マッピングを実行するのに使用可能であり、前記読取マッピング・テーブルおよび前記書込マッピング・テーブルを使用して、データのブロックのコピー・オン・ライトが単一の書込動作を使用して達成可能である、論理
を含む、製造品。
（５０）製造品であって、
複数のクライアントを有するクライアント・サーバ環境内でファイルのコピー・オン・ライトを実施するコンピュータ可読プログラム・コード手段を有する少なくとも１つのコンピュータ使用可能媒体
を含み、前記コンピュータ可読プログラム・コード手段が、
前記クライアント・サーバ環境の複数のクライアントを使用してファイルのコピー・オン・ライトを実行する論理
を含み、前記実行する論理が、
（ｉ）前記複数のクライアントの第１クライアントによって、前記ファイルのデータの少なくとも１つのブロックのコピー・オン・ライトを実行する論理と、
（ｉｉ）前記複数のクライアントの第２クライアントによって、前記ファイルのデータの少なくとも１つの他のブロックのコピー・オン・ライトを実行する論理と
を含む、製造品。
（５１）前記論理（ｉ）が、前記第１クライアントによって、単一の書込動作を使用して前記ファイルのデータの前記少なくとも１つのブロックのコピー・オン・ライトを実行する論理を含み、前記論理（ｉｉ）が、前記第２クライアントによって、単一の書込動作を使用してデータの前記少なくとも１つの他のブロックのコピー・オン・ライトを実行する論理を含む、上記（５０）に記載の製造品。
（５２）前記クライアント・サーバ環境のファイルシステム・サーバが、前記ファイルを含む少なくとも１つの共用記憶装置に関連し、前記ファイルシステム・サーバが、前記少なくとも１つの共用記憶装置に保管された前記ファイルに関するマッピング・テーブルを維持し、前記論理（ｉ）が、前記第１クライアントによって、前記ファイルシステム・サーバから前記ファイルに関する読取マッピング・テーブルおよび書込マッピング・テーブルを入手し、前記ファイルのデータの前記少なくとも１つのブロックの前記コピー・アンド・ライトを実行する際に前記読取マッピング・テーブルおよび前記書込マッピング・テーブルを使用する論理を含み、前記論理（ｉｉ）が、前記第２クライアントによって、前記ファイルシステム・サーバから前記ファイルに関する前記読取マッピング・テーブルおよび前記書込マッピング・テーブルを入手し、前記ファイルのデータの前記少なくとも１つの他のブロックの前記コピー・アンド・ライトを実行する際に前記読取マッピング・テーブルおよび前記書込マッピング・テーブルを使用する論理を含む、上記（５１）に記載の製造品。
（５３）製造品であって、
クライアント・サーバ環境内でファイルのコピー・オン・ライトを容易にするコンピュータ可読プログラム・コード論理を有する少なくとも１つのコンピュータ使用可能媒体
を含み、前記コンピュータ可読プログラム・コード論理が、
前記クライアント・サーバ環境の共用記憶ユニットに保管されたファイルのコピー・オン・ライトの実施をファイルシステム・サーバから制御する論理であって、前記制御する論理が、前記クライアント・サーバ環境の第１クライアントが前記ファイル内のデータの部分をコピー・オン・ライトできるようにする論理と、前記クライアント・サーバ環境の第２クライアントが前記ファイル内の前記データの異なる部分をコピー・オン・ライトできるようにする論理とを含み、前記ファイルシステム・サーバが、前記ファイルの分散コピー・オン・ライトの実行を制御し、容易にする、制御する論理
を含む、製造品。
（５４）前記制御する論理が、前記ファイルのコピー・オン・ライトを開始するために前

記ファイルシステム・サーバによって前記ファイルを制御する論理を含み、前記開始することが、前記コピー・オン・ライトを実行するのに使用される前記ファイルに関する少なくとも１つのマッピング・テーブルを更新することを含む、上記（５３）に記載の製造品。

（５５）前記ファイルシステム・サーバで、前記ファイルに関する読取マッピング・テーブルおよび書込マッピング・テーブルを維持する論理をさらに含み、前記読取マッピング・テーブルおよび前記書込マッピング・テーブルが、前記コピー・オン・ライトの実行に使用される、上記（５３）に記載の方法。

【図面の簡単な説明】

【０１０７】

【図１】本発明の１つまたは複数の態様が組み込まれ、使用されるコンピューティング環境の一実施形態を示す図である。

【図２】範囲またはオフセットが仮想／相対ブロック番号に変換される、アプリケーションまたはプロセスに対するファイル表現（２００）の一例を示す図である。

【図３】図２の仮想／相対ブロック番号が、ファイルシステムの１つまたは複数の記憶ユニット内の物理ブロック・アドレスにマッピングされる、ファイルシステム・マッピング・テーブル（３００）を示す図である。

【図４】ファイルシステムの記憶ユニット内の図３の物理ブロック・アドレスの配置を示す図である。

【図５】オフセット５０００から開始され、ファイルシステムの記憶ユニットの物理ブロックＤ内に含まれる４バイトのデータの読取の例を示す図である。

【図６】本発明の態様による書込動作およびコピー・オン・ライト動作の一実施形態を示す流れ図である。

【図７】図６の論理に従う書込動作で使用されるデータのブロックを記憶ユニットから論理バッファまたはキャッシュに読み取る読取処理の例を示す流れ図である。

【図８】図６の論理に従う書込動作で使用されるデータの修正されたブロックを記憶ユニットに書き込む書込処理の例を示す流れ図である。

【図９】本発明の態様による、ファイルのコピー・オン・ライトに使用される読取マッピング・テーブル（９００）および書込マッピング・テーブル（９１０）の例を示す図である。

【図１０】本発明の態様による、図６の論理によるコピー・オン・ライト動作に使用されるデータ・ブロック読取処理の一実施形態を示す流れ図である。

【図１１】本発明の態様による、図６の論理によるコピー・オン・ライト動作に関する修正済みデータ・ブロック書込処理の一実施形態を示す流れ図である。

【図１２】本発明の態様による、ファイルのコピー・オン・ライトに使用される読取マッピング・テーブル（１２００）および書込マッピング・テーブル（１２１０）のもう１つの例を示す図である。

【図１３】本発明の１つまたは複数の態様が組み込まれ、使用されるコンピューティング環境のもう１つの例を示す図である。

【図１４】本発明の態様による、たとえば図１３のクライアント・サーバ環境の、ファイルの少なくとも１ブロックのデータのコピー・オン・ライトを実行するクライアント１の一実施形態を示す流れ図である。

【図１５】本発明の態様による、たとえば図１３のクライアント・サーバ環境の、ファイルの少なくとも１つの他のブロックのデータのコピー・オン・ライトを実行するクライアント２の一実施形態を示す流れ図である。

【図１６】本発明の態様による、コピー・オン・ライト中に使用されるファイルに関する読取マッピング・テーブル（１６００）および書込マッピング・テーブル（１６１０）のもう１つの例を示す図である。

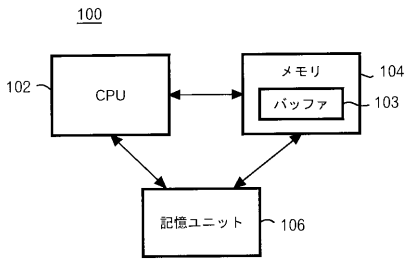
【図１７】本発明の態様による、コピー・オン・ライトを使用するデータ保存アプリケーションの一例を示す流れ図である。

【符号の説明】

【０１０８】

１００　コンピューティング環境
１０２　中央処理装置
１０４　メモリ
１０３　バッファまたはキャッシュ領域
１０６　記憶ユニットまたは記憶装置
２００　ファイル表現
３００　ファイルシステム・マッピング・テーブル
４００　記憶ユニット
９００　読取マッピング・テーブル
９１０　書込マッピング・テーブル
１２００　更新された読取マッピング・テーブル
１２１０　書込マッピング・テーブル
１３００　環境
１３１０　クライアント１
１３２０　クライアント２
１３３０　ファイルシステム・サーバ
１３４０　ストレージ・エリア・ネットワーク
１３５０　ストレージ・プール
１６００　読取マッピング・テーブル
１６１０　書込マッピング・テーブル

【図１】



【図２】

200　ファイル表現

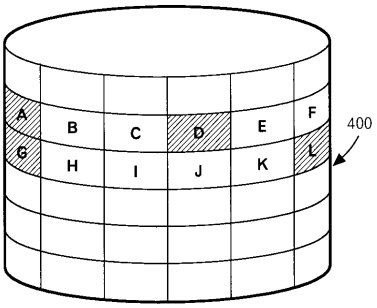
範囲／オフセット	仮想／相対ブロック番号
0－4K	1
4－8K	2
8－12K	3
12－16K	4

【図３】

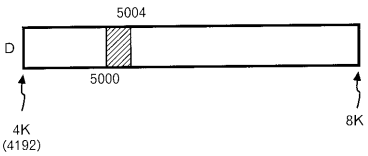
300　ファイル・システム・マッピング・テーブル

仮想／相対ブロック番号	物理ブロック・アドレス
1	A
2	D
3	G
4	L

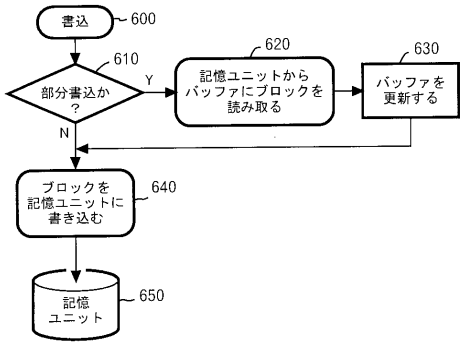
【図４】



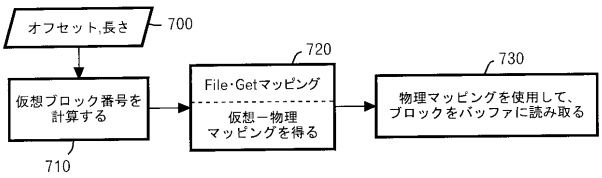
【図５】



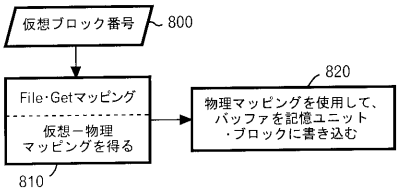
【図６】



【図７】



【図８】



【図１２】

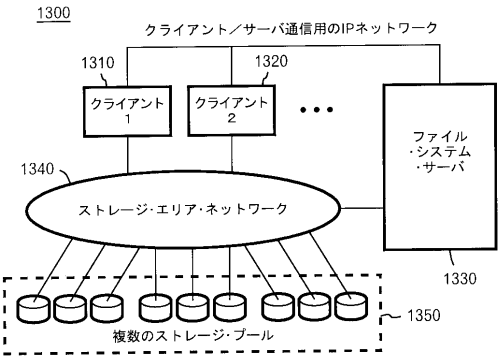
1200 読取 マッピング･テーブル

仮想ブロック番号	物理ブロック番号
1	A
2	X
3	G
4	L

1210 書込 マッピング･テーブル

仮想ブロック番号	物理ブロック番号
1	W
2	X
3	Y
4	Z

【図１３】



【図９】

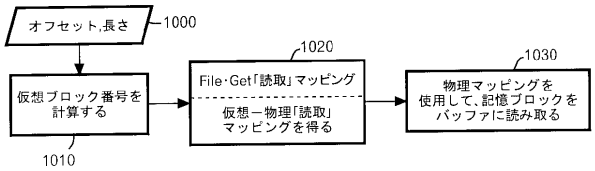
900 読取 マッピング･テーブル

仮想ブロック番号	物理ブロック番号
1	A
2	D
3	G
4	L

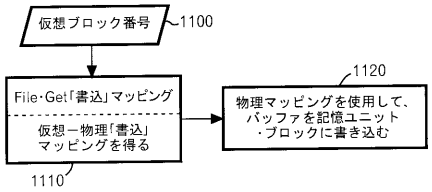
910 書込 マッピング･テーブル

仮想ブロック番号	物理ブロック番号
1	W
2	X
3	Y
4	Z

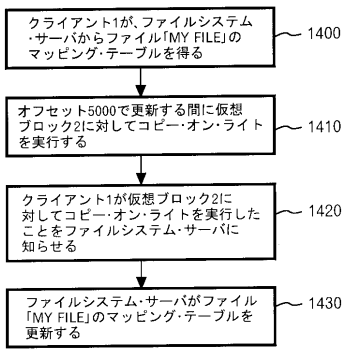
【図１０】



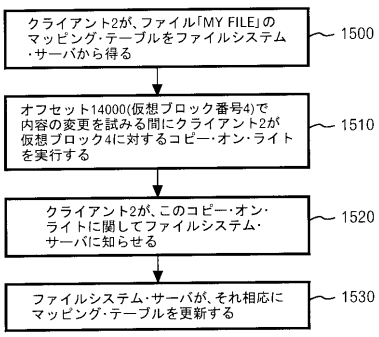
【図１１】



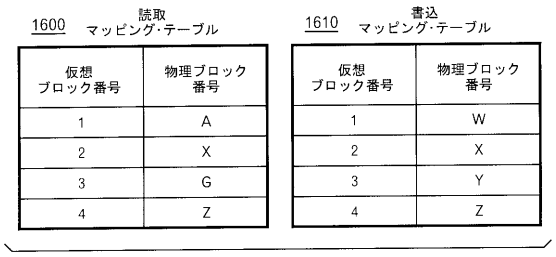
【図１４】



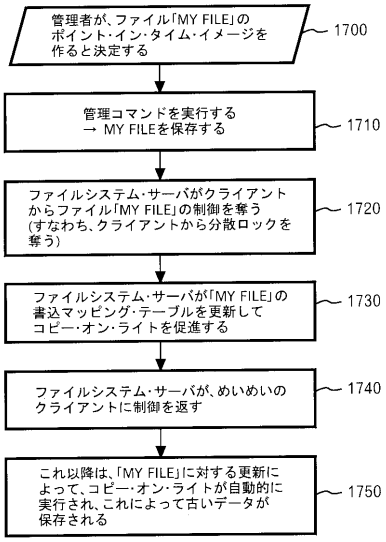
【図１５】



【図１６】


1600
読取
マッピング・テーブル
仮想
ブロック番号
物理ブロック
番号
1
A
2
X
3
G
4
Z
1610
書込
マッピング・テーブル
仮想
ブロック番号
物理ブロック
番号
1
W
2
X
3
Y
4
Z

【図１７】


管理者が、ファイル「MY FILE」の
ポイント・イン・タイム・イメージを
作ると決定する
1700
管理コマンドを実行する
→ MY FILEを保存する
1710
ファイルシステム・サーバがクライアント
からファイル「MY FILE」の制御を奪う
(すなわち、クライアントから分散ロックを
奪う)
1720
ファイルシステム・サーバが「MY FILE」の
書込マッピング・テーブルを更新して
コピー・オン・ライトを促進する
1730
ファイルシステム・サーバが、めいめいの
クライアントに制御を返す
1740
これ以降は、「MY FILE」に対する更新に
よって、コピー・オン・ライトが自動的に
実行され、これによって古いデータが
保存される
1750

フロントページの続き

(72)発明者 ラジャゴパル・アナンタナラヤナン
アメリカ合衆国９５０３５ カリフォルニア州ミルピータス サーク・コート ５２２
(72)発明者 ラルフ・エー・ベッカーツェンディ
アメリカ合衆国９５０３３ カリフォルニア州ロス・ガトス サンセット・リッジ・ロード ７３５
(72)発明者 ロバート・エム・リース
アメリカ合衆国９５０３３ カリフォルニア州ロス・ガトス グリーンウッド・ドライブ １７９６３
(72)発明者 ランダル・シー・バーンズ
アメリカ合衆国２００１１ ワシントンＤ．Ｃ． ウェブスター・ストリート ノース・ウェスト １５２０
(72)発明者 ダレル・ディー・イー・ロング
アメリカ合衆国９５０７３ カリフォルニア州ソーケル ワイルダー・ドライブ ５０３０
(72)発明者 ヴェンカテスワラオ・ジュジュリ
アメリカ合衆国９７００６ オレゴン州ビーヴァートン ノースウェスト・トゥーソン・ストリート １７１７７
(72)発明者 デーヴィッド・エム・ウルフ
アメリカ合衆国９７２２９ オレゴン州ポートランド ノースウェスト・スターク・コート ９１９５
(72)発明者 ジェイソン・シー・ヤング
アメリカ合衆国９７２２９ オレゴン州ポートランド ノースウェスト・カントリリッジ・ドライブ １６９７２

Ｆターム(参考) 5B065 BA01 CC01 CC03 EA33
5B082 FA05